KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ SISTEMŲ INŽINERIJOS KATEDRA



Deividas Kančelkis

Lokaliai progresyvaus vaizdų kodavimo metodo realizacija ir tyrimas

Magistro darbas

Darbo vadovas
prof. J. Valantinas, TMK

Kaunas, 2008

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ SISTEMŲ INŽINERIJOS KATEDRA

TVIRTINU
Katedros vedėjas
(parašas) prof. E.Bareiša
2008 05

Lokaliai progresyvaus vaizdų kodavimo metodo realizacija ir tyrimas

Magistro darbas

Vadovas
prof. J. Valantinas, TMK
(parašas) 2008 05

Recenzentas
doc. dr.
(parašas) 2008 05

Autorius
D. Kančelkis, IFM–2/2
(parašas) 2008 05 26

Kaunas, 2008

TURINYS

SANTRAUKA 10
1. ĮVADAS 11
2. SKAITMENINIŲ VAIZDŲ SUGLAUDINIMO SPEKTRINĖJE SRITYJE METODŲ APŽVALGA 14
2.1 Diskrečiosios transformacijos ir jų greito skaičiavimo algoritmai 15
2.1.1 Diskrečioji Furjė transformacija (DFT) 15
2.1.2 Volšo ir Adamaro diskrečioji transformacija (VAT) 19
2.1.3 Diskrečioji kosinusinė transformacija (DKT) 21
2.1.4 Diskrečioji Haaro transformacija 22
2.2 Vaizdų suglaudinimo spektrinėje srityje metodų apžvalga 25
2.2.1 Dispersiniai kriterijai 25
2.2.2 JPEG standartas 26
2.2.3 Hiperboliniai vaizdų filtrai 28
2.2.4 Fraktalinės vaizdų suglaudinimo procedūros 29
2.3 Vaizdų suglaudinimo metodai be informacijos praradimo 30
2.3.1 RLE kodavimo metodas 30
2.3.2 LZW kodavimo metodas 31
2.4 Dekoduoto vaizdo kokybės įvertinimas 31
2.5 Progresyviojo vaizdų kodavimo idėja – EZW algoritmas 32
2.6 Diskrečiosios Haaro (bangelių) transformacijos savybių bei skaičiavimo algoritmų analizė 35
2.6.1 Specifinės HT koeficientų savybės – lokalizavimas erdvėje 35
2.6.2 Specifinės HT koeficientų savybės – nulinių medžių egzistavimas 35
2.7 Naujas diskrečiojo Haaro spektro apskaičiavimo vaizdo fragmentams algoritmas 36
2.7.1 Vienmatis atvejis 36
2.7.2 Dvimatis atvejis 38
3. LOKALIAI PROGRESYVAUS VAIZDŲ KODAVIMO PROGRAMINĖS ĮRANGOS PROBLEMOS IR JŲ SPRENDIMO ANALIZĖ 39
3.1 Projekto tikslas 39
3.2 Užduotis 41
3.3 Sistemos įgyvendinimo etapai 41
3.4 Problemos sprendimas pasaulyje ir Lietuvoje 42
3.5 Pagrindinės sistemos projektavimo, programavimo inžinerijos problemos, jų sprendimai 44
3.5.1 Objektiškai orientuoto programavimo problema ir jos sprendimas 44
4. PROGRAMINĖS ĮRANGOS TECHNINĖS/PROJEKTINĖS DOKUMENTACIJOS ESMINIAI ASPEKTAI 46
4.1 Projekto apribojimai 46
4.2 Pagrindinės sistemos funkcijos 46
4.3 Sistemos sudėtis 48
4.4 Sistemai keliami reikalavimai 49
4.4.1 Funkciniai reikalavimai 49
4.4.2 Nefunkciniai reikalavimai 50
5. TYRIMAS 51
5.1 Kokybės vertinimo procesas 51
5.1.1 Tinkamumas 51
5.1.2 Tikslumas 51

5.1.3 Apsauga 51
5.1.4 Išmokstamumas 52
5.1.5 Veikimas 52
5.1.6 Laikinė elgsena 52
5.1.7 Elgsena resursų atžvilgiu 52
5.1.8 Analizuojamumas 52
5.1.9 Keičiamumas 52
5.1.10 Stabilumas 52
5.1.11 Testuojamumas 52
5.1.12 Prisitaikomumas 53
5.1.13 Įdiegiamumas 53
5.2 Vertinimo rezultatai 53
5.3 Sistemos tobulinimas 54
6. EKSPERIMENTO REZULTATAI 54
6.1 Eksperimento turinys 54
6.2 Skaitmeninio vaizdo kodavimas 55
6.2.1 Kodavimo realizacija 55
6.2.2 Kodavimo metodų vykdymo laikų matavimas 56
6.2.3 Kodavimo operacijos spartinimas 59
6.2.4 Tikslinis kodavimo metodų modifikavimas 61
6.3 HT spektro radimo procedūros teorinis efektyvumo įvertinimas 63
6.3.1 Vienmatis atvejis 63
6.3.2 Dvimatis atvejis 64
6.4 HT spektro radimo procedūros praktinis efektyvumo įvertinimas 65
6.4.1 HT spektro radimo laiko sąnaudų priklausomybė nuo fragmento vietos vaizde 65
6.4.2 HT spektro radimo laiko sąnaudų priklausomybė nuo fragmento dydžio 66
6.4.3 HT spektro radimo laiko sąnaudų priklausomybė nuo vaizdo dydžio 67
7. IŠVADOS 68
LITERATŪRA 68
SUMMARY 69
SANTRUMPŲ IR TERMINŲ ŽODYNĖLIS 70
PRIEDAI 70
1. PRIEDAS. TIKSLINIO KODAVIMO METODŲ MODIFIKAVIMO REZULTATAI 512X512 VAIZDAMS 70
2. PRIEDAS. TIKSLINIO KODAVIMO METODŲ MODIFIKAVIMO REZULTATAI 1024X1024 VAIZDAMS 72
3. PRIEDAS. APIE LOKALIAI PROGRESYVŲJĮ DVIMAČIŲ SKAITMENINIŲ VAIZDŲ KODAVIMĄ 74
1. Įvadas 74
2. Diskrečiosios Haaro transformacijos (HT) apskaičiavimas vaizdo fragmentams 74
3. Lokaliai progresyvaus vaizdų kodavimo idėja 76
4. Išvados 78

PAVEIKSLĖLIŲ SĄRAŠAS

2.2 pav. GFT „signalinis“ grafas, kai $N = 16$ 17
2.3 pav. VAT_h skaičiavimo algoritmų „signaliniai“ grafai: (a) „rankinis“; (b) „Cooley-Tukey“ tipo 20
2.4 pav. Greitų HT ir AHT transformacijų apskaičiavimo algoritmų „signaliniai“ grafai: (a) HT; (b) AHT ($N = 2^3 = 8$; $c_r = c_{r-1} \cdot \sqrt{2}$, $r = 2, 3$; $c_1 = 1$) 24
2.5 pav. Zigzaginis kvantuotų spektrinių DKT koeficientų nuskaitymas 27
2.6 pav. Hiperbolinio vaizdų filtro veikimo schema 29
2.7 pav. Fraktalinio vaizdų kodavimo schema – būtinos vaizdų panašumo sąlygos taikymas 30
2.8 pav. Spektro skenavimo schema (EZW): (a) simbolio priskyrimo spektriniam koeficientui schema; (b) spektro skenavimo tvarka (Morton seka) 34
2.9 pav. Geometrinė spektrinių HT koeficientų interpretacija: (a) su koeficientu $Y(k)$ susijusio medžio struktūra; (b) koeficientas $Y(3)$ ir jį atitinkantis medis (paryškintas; $N = 16$); (c) vaizdo fragmentai X_{k^*}, su kuriais susiję medžio šaknis $Y(3)$ bei medžio viršūnės $Y(k^*)$ ($N = 16$) 36
2.10 pav. Haaro spektro savybių grafinė interpretacija: (a) vaizdas $[X(m_1, m_2)]$, $N = 16$; spektrinis koeficientas $Y(2, 3)$ susijęs su vaizdo fragmentu $X_{2,3}$; (b) koeficientui $Y(2, 3)$ priskirtas medis ($\Im_2 = \Im_2(1) \bigcup \Im_2(2) = \{4, 5, 8, 9, 10, 11\}$, $\Im_3 = \Im_3(1) \bigcup \Im_3(2) = \{6, 7, 12, 13, 14, 15\}$; čia $\Im_{k_i}(t) = \{2^t \cdot k_i, 2^t \cdot k_i + 1, ..., 2^t \cdot (k_i + 1) - 1\}$, su visais $t = 1,2,...,s_i - 1$, $i = 1,2$.) 37
3.1 pav. Progresyvus (c, e, g, i) ir lokaliai progresyvus (d, f, h, j) vaizdų kodavimas: (a) originalas „*cameraman.bmp*“ 512x512, 256 KB; (b) vaizdo diskretusis Haaro spektras; (c) suspaustas vaizdas $\beta = 40$, 6.4 KB; (d) suspaustas vaizdas $\beta = 20$, 12.8 KB; (e) suspaustas vaizdas $\beta = 40$, fragmentas $\beta = 20$, 7.2 KB; (f) suspaustas vaizdas $\beta = 10$, 25.6 KB; (f) suspaustas vaizdas $\beta = 40$, fragmentas $\beta = 10$, 8.8 KB; (g) suspaustas vaizdas $\beta = 5$, 51.2 KB; (h) suspaustas vaizdas $\beta = 40$, fragmentas $\beta = 5$, 12 KB; (i) suspaustas vaizdas $\beta = 2$, 128 KB; (j) suspaustas vaizdas $\beta = 40$, fragmentas $\beta = 2$, 18.4 KB; 41
3.2 pav. JPEG formato failo dydžio priklausomybė nuo vaizdo kokybės 44
3.3 pav. OOP sudėtingumo įtaka sistemos (HT, EZW modeliams) darbo našumui 46
4.1 pav. Skaitmeninio vaizdo apdorojimo (kodavimo, dekodavimo, detalizavimo) veiksmų seka 47
4.2 pav. Sistemos panaudos atvejų modelis 48
6.1 pav. Eksperimento rezultatai; (a) originalus vaizdas; (b) HT, EZW, kai $t_{min} = 1$, $\beta = 16$; (c) HF, kai $\beta = 16$; (d) JPEG, kai $\beta = 16$; (e) HT, EZW, kai $t_{min} = 2$, $\beta = 70$; (f) HF, kai $\beta = 70$; (g) JPEG, kai $\beta = 70$; (h) HT, EZW, kai $t_{min} = 4$, $\beta = 222$; (i) HF, kai $\beta = 222$; (j) JPEG, kai $\beta = 88$. 56
6.2 pav. Vaizdo 256x256 kodavimo ir dekodavimo operacijų trukmės grafikas pagal suspaudimo laipsnį 57
6.3 pav. Vaizdo 512x512 kodavimo ir dekodavimo operacijų trukmės grafikas pagal suspaudimo laipsnį 58
6.4 pav. Vaizdo 1024x1024 kodavimo ir dekodavimo operacijų trukmės grafikas pagal suspaudimo laipsnį 58
6.5 pav. Vaizdo spektras ir reikšmingi spektro koeficientai; (a) 256x256 raškos vaidas; (b) 1024x1024 raiškos vaizdas 59
6.6 pav. HT skaičiavimo trukmės (greita HT, PROMOL kalbos komponentais grįsta HT, tiesioginė HT) 60

6.7 pav. Tiesioginio ir atvirkštinio EZW algoritmo vykdymo greičių priklausomybė nuo spektro dydžio (*N*x*N*)..... 61
6.8 pav. Eksperimento rezultatai: spektro (256x256) reikšmių kėlimo daugikliu *c* įtaką atkuriamam vaizdui; (a) originalas; (b) *c* = 1, PSNR = 24,062 dB (sistema nemodifikuota); (c) *c* = 2, PSNR = 26,82 dB; (d) *c* = 5, PSNR = 32,095 dB; (e) *c* = 10, PSNR = 36,308 dB; (f) *c* = 20, PSNR = 39,084 dB; (g) *c* = 50, PSNR = 39,323 dB..... 62
6.9 pav. Spektro daugiklio įtaka vaizdo kodavimo ir dekodavimo trukmei..... 63
6.10 pav. Tiesioginės ir netiesioginės procedūrų operacijų palyginimas, kai *n* = 11 65
6.11 pav. Tiesioginės ir netiesioginės procedūrų laiko sąnaudų įverčių priklausomybė nuo fragmento dydžio 66
6.12 pav. Tiesioginės ir netiesioginės procedūrų laiko sąnaudų įverčių priklausomybė nuo fragmento dydžio (logaritminė skalė)..... 67
1.1 pav. Eksperimento rezultatai: spektro (512x512) reikšmių kėlimo daugikliu *c* įtaką atkuriamam vaizdui; (a) originalas „*laguna.bmp*“, 256 KB; (b) *c* = 1, PSNR = 24,321 dB (sistema nemodifikuota); (c) *c* = 2, PSNR = 26,342 dB; (d) *c* = 5, PSNR = 29,649 dB; (e) *c* = 10, PSNR = 32,681 dB; (f) *c* = 20, PSNR = 36,634 dB; (g) *c* = 50, PSNR = 41,536 dB..... 71
1.2 pav. Spektro daugiklio įtaka vaizdo kodavimo ir dekodavimo trukmei..... 72
2.1 pav. Eksperimento rezultatai: spektro (1024x1024) reikšmių kėlimo daugikliu *c* įtaką atkuriamam vaizdui; (a) originalas „*tower.bmp*“, 1 MB; (b) *c* = 1, PSNR = 22,984 dB (sistema nemodifikuota); (c) *c* = 2, PSNR = 24,863 dB; (d) *c* = 5, PSNR = 27,948 dB; (e) *c* = 10, PSNR = 30,542 dB; (f) *c* = 20, PSNR = 33,587 dB; (g) *c* = 50, PSNR = 38,751 dB..... 73
2.2 pav. Spektro daugiklio įtaka vaizdo kodavimo ir dekodavimo trukmei..... 74
2.1 pav. HT skaičiavimo algoritmo signalinis grafas (*N* = 8)..... 75
2.2 pav. Haaro spektro savybių grafinė interpretacija: (a) vaizdas [$X(m_1, m_2)$], $N = 16$, spektro koeficientas $Y(2, 3)$ susijęs su vaizdo fragmentu $X_{2,3}$; (b) koeficientui $Y(2, 3)$ priskirtas medis ($\Im_2 = \Im_2(1) \cup \Im_2(2) = \{4, 5, 8, 9, 10, 11\}$, $\Im_3 = \Im_3(1) \cup \Im_3(2) = \{6, 7, 12, 13, 14, 15\}$)..... 76
3.1 pav. Medžio struktūros ir jų skenavimo trajektorijos: (a) medžių struktūros (kiekvienas spektrinis koeficientas turi 4 spektrinius koeficientus aukštesniuose dažniuose); (b) Morton skenavimo kreivė 76
3.2 pav. Lokaliai progresyvaus skaitmeninių vaizdų kodavimo rezultatai: (a) originalus vaizdas *tester.bmp* 512 × 512; (b) atkurtas „grubus“ vaizdas, kai $t_{min} = 8$; (c) progresyvaus kodavimo rezultatas (patikslintos detalės lokalizuotame vaizdo fragmente $X_{2,5}$)..... 77
3.3 pav. Lokaliai progresyvaus skaitmeninių vaizdų kodavimo rezultatai: (a) originalus vaizdas *laguna.bmp* 512 × 512; (b) atkurtas „grubus“ vaizdas, kai $t_{min} = 8$; (c) progresyvaus kodavimo rezultatas (patikslintos detalės lokalizuotame vaizdo fragmente $X_{6,5}$)..... 78

IŠRAIŠKŲ SĄRAŠAS

(2.1) ... 14
(2.2) ... 14
(2.3) ... 14
(2.4) ... 15
(2.5) ... 15
(2.6) ... 16
(2.7) ... 16
(2.8) ... 16
(2.9) ... 16
(2.10) ... 18
(2.11) ... 18
(2.12) ... 18
(2.13) ... 19
(2.14) ... 19
(2.15) ... 19
(2.16) ... 19
(2.17) ... 21
(2.18) ... 21
(2.19) ... 21
(2.20) ... 21
(2.21) ... 22
(2.22) ... 22
(2.23) ... 23
(2.24) ... 23
(2.25) ... 23
(2.26) ... 23
(2.27) ... 23
(2.28) ... 25
(2.29) ... 25
(2.30) ... 26
(2.31) ... 27
(2.32) ... 27
(2.33) ... 28
(2.34) ... 28
(2.35) ... 32
(2.36) ... 32
(2.37) ... 37
(2.38) ... 37
(2.39) ... 37
(2.40) ... 37
(2.41) ... 38
(2.42) ... 38
(2.43) ... 39
(2.44) ... 39
(2.45) ... 39
(6.1) ... 64
(6.2) ... 64

(6.3)..... 64
(6.4)..... 64

LENTELIŲ SĄRAŠAS

2.1 lentelė. Svorių *W* nustatymo lentelė 18
3.1 lentelė. 2006 – 2007 m.m. darbų tvarkaraštis 42
3.2 lentelė. 2007 – 2008 m.m. darbų tvarkaraštis 42
3.3 lentelė. JPEG kodavimo standarto rezultatai 43
5.1 lentelė. Programinės įrangos kokybės kriterijų įvertinimas 53
6.1 lentelė. HT ir EZW metodų rezultatai 55
6.2 lentelė. HT ir EZW metodų vykdymo laikų rezultatai 56
6.3 lentelė. Spektro daugiklio įtaka koduojamo vaizdo apimčiai 62
6.4 lentelė. Spektro daugiklio įtaka kodavimo trukmei 63
6.5 lentelė. Tiesioginės ir netiesioginės procedūrų operacijų skaičius, jų palyginimas, kai $n = 11$ 64
6.6 lentelė. Tiesioginės ir netiesioginės procedūrų operacijų skaičius, jų palyginimas, kai $n = 11$ 65
6.7 lentelė. Diskretaus Haaro spektro radimo procedūrų laiko sąnaudų įverčių priklausomybė nuo fragmento vietos vaizde 65
6.8 lentelė. Diskretaus Haaro spektro radimo procedūrų laiko sąnaudų įverčių priklausomybė nuo fragmento dydžio 66
6.9 lentelė. Diskretaus Haaro spektro radimo procedūrų laiko sąnaudų įverčių priklausomybė nuo vaizdo dydžio 67
1.1 lentelė. Spektro daugiklio įtaka koduojamo vaizdo apimčiai 71
1.2 lentelė. Spektro daugiklio įtaka kodavimo trukmei 71
2.1 lentelė. Spektro daugiklio įtaka koduojamo vaizdo apimčiai 73
2.2 lentelė. Spektro daugiklio įtaka kodavimo trukmei 73

SANTRAUKA

Skaitmeniniai vaizdai yra plačiai naudojami kompiuterių taikomosiose programose. Nesuspaustų skaitmeninių vaizdų laikymui atmintyje reikia žymiai daugiau talpos ir spartesnio (didesnio pralaidumo) tinklo galimybių jų perdavimui tinklu. Efektyvūs vaizdų suspaudimo sprendimai tampa kritiškesni dėl pastaruoju metu didelio duomenų augimo intensyvumo, multimedijos pagrindu kuriamų tinklo programų.

Šiame darbe pristatoma nauja lokaliai progresyvaus vaizdų kodavimo idėja (procedūra). Procedūra grindžiama specifinėmis Haaro bangelių savybėmis bei progresyvųjį vaizdų kodavimą realizuojančiu EZW algoritmu. Taip pat apžvelgiamos kitų diskrečiųjų transformacijų ypatybės, praktinis jų pritaikymas pramoninėje, mokslinėje srityse. Preliminarūs eksperimento rezultatai rodo, jog diskrečioji Haaro transformacija ir progresyvus vaizdų glaudinimo algoritmas EZW skaitmeninių vaizdų spaudimui nėra efektyvus. Efektyvumo didinimui pasirinktas tikslinis vaizdo spektro koeficientų modifikavimas, kas leido pasiekti kur kas geresnių kodavimo rezultatų.

1. ĮVADAS

Esame kompiuterių ir komunikavimo technologijų revoliucijos liudininkai. Matome, kaip telekomunikacijų pramonė išplito visame pasaulyje, analoginį signalą pakeisdama skaitmeniniu, kas, savo ruožtu, leido ištobulėti šiandieninei labai galingai interneto technologijai. Perėjimas nuo analoginio prie skaitmeninio signalo visame pasaulyje suteikė daug įvairių galimybių. Telekomunikacijos, internetas, skaitmeninės pramogos, sudėtingi skaičiavimai apskritai tampa mūsų kasdienio gyvenimo dalimi. Šiandien jau kalbame apie skaitmeninius tinklus, skaitmeninių statinių vaizdų, filmų, video, TV, balso, skaitmeninės bibliotekos pristatymus. Neabejotinai, skaitmeninis signalas geresnis (tinkamesnis) nei analoginis dublikatas duomenų apdorojimui, manipuliavimui, laikymui, atkūrimui ir perdavimui dideliais atstumais (netgi aplink Žemės rutulį per komunikacijos tinklus). Pastaraisiais metais stipriai pažengta į priekį statinių vaizdų, video, grafikos, kalbos ir audio signalų apdorojime, kas leido išspręsti įvairius taikomuosius uždavinius. Daugialypė (multimedija) informacija, apimanti vaizdą, video, audio, kalbą, tekstą ir pan. turi potencialą tapti kitu duomenų tipu. Telekomunikacija daugiau nėra platforma, skirta P2P[1] tipo komunikavimui tarp dviejų žmonių. Komunikacijų reikalavimai sparčiai auga multimedijos duomenų priėjimui telekomunikacijų tinklais ir internetu. Tam, kad suvaldytumėme šį multimedijos duomenų panaudojimo plitimą, svarbu, kad multimedijos duomenų pateikimas ir kodavimas būtų standartas visoms skirtingoms platformoms ir programoms. Statiniai vaizdai ir video duomenys sudaro žymią (reikšmingą) multimedijos duomenų dalį ir užima didelę plačiajuosčio komunikacijos tinklo dalį. Todėl efektyvių vaizdų suglaudinimo (suspaudimo) metodų kūrimas bei vystymas tampa svarbiu iššūkiu tiek mokslui, tiek pramonei.

Nepaisant daugelio skaitmeninių signalų pranašumų lyginant su analoginiais signalais, jų laikymui (saugojimui) ir perdavimui reikia labai didelio bitų kiekio. Pavyzdžiui, aukštos kokybės audio signalo skaitmeniniam pateikimui ir laikymui atmintyje reikia apytiksliai 1,5 Mb/s. Televizijos lygio, nedidelės sklaidos (kuomet kadras sudarytas iš 640 x 480 vaizdo elementų) 30 spalvotų kadrų/s kokybės atveju reikia daugiau kaip 210 Mb/s atminties. Todėl, pervestam į skaitmeninę formą vienos valandos spalvotam kino filmui reikia maždaug 95 GB talpos. Pramonėje įsigalinčiai aukštos sklaidos (didelio tikslumo) televizijai, kitaip dar žinomai kaip HDTV, kuomet kadras sudarytas iš 1280 x 60 vaizdo elementų bei pateikiama 60 kadrų/s sparta,

[1] Angl. peer-to-peer — P2P tinkle nėra kliento ir serverio sąvokų – abu susijungę vartotojai yra lygiaverčiai, kiekvienas veikia ir kaip klientas ir kaip serveris

reiktų žymiai didesnės talpos. Tokiu atveju į skaitmeninį formatą (skaitmeninę formą) pervestas vienos valandos spalvotas HDTV kokybės kino filmas pareikalaus apie 560 GB atminties. Į skaitmeninę formą pervesta 14 x 17 colių rentgenograma, nuskenuota 70 μm ilgio bangomis, užims beveik 45 MB atminties. Šių skaitmeninių signalų perdavimas riboto pralaidumo komunikacijos kanalais yra dar didesnis iššūkis. Nors atminties kaina, dėka žymių (reikšmingų) mikroelektronikos ir atminties technologijų pažangos, per pastarąjį dešimtmetį drastiškai sumažėjo, tačiau duomenų laikymo ir naudojimo apimtys aplikacijose auga dar sparčiau.

Įdomu tai, kad dauguma sensorinių signalų, tokių kaip statiniai vaizdai, video ar balso, apskritai, savyje talpina žymius (užčiuopiamus) kiekius nereikšmingos ir perteklinės informacijos, kurios žmogaus suvokiamoji (priimamoji) sistema neužfiksuoja. Šią sistemą atitiktų matymo ir klausos gebėjimai. Pavyzdžiui, kaimyniniai vaizdo elementai glodžioje realaus pasaulio vaizdo srityje yra labai panašūs ir mažas kaimyninių vaizdo elementų reikšmių pokytis nėra pastebimas žmogaus akiai. Nuoseklūs (einantys iš eilės) video kadrai stacionarioje ar lėtai besikeičiančioje scenoje yra labai panašūs, o dalis regimosios informacijos yra perteklinė. Kai kurie audio duomenys, esantys už žmogaus girdimo dažnio diapazono, yra nenaudingi visais praktiniais aspektais. Šis faktas sako, kad tokiuose signaluose yra duomenų, kurie žmogaus priimamai (jutiminei) sistemai nėra suvokiami. Tai vadinama jutiminiu (suvokimo) pertekliumi.

Duomenų suglaudinimas (suspaudimas) – tai metodas (technika), pašalinantis duomenų pertekliškumą, siekiant sumažinti duomenų atminties poreikį. Tai pagrindinis duomenų suspaudimo privalumas. Tai patrauklus būdas sumažinti komunikavimo kainas perduodant dideles duomenų apimtis, naudojantis tolimo nuotolio ryšio linijomis. Kita vertus, tai padeda žymiai sumažinti komunikacijų kainas dėl duomenų normos sumažėjimo. Dėl duomenų normos sumažėjimo, duomenų suspaudimas didina multimedijos kokybę perduodant riboto pralaidumo komunikacijų kanalais. Dėl ženklių vaizdo suspaudimo technologijų, vienas 6 MHz televizijos transliacijos kanalas gali perduoti HDTV signalus be papildomo plačiajuosčio tinklo srauto poreikio. Dėl suspaudimo technologijų sumažintos duomenų normos, vaizdų ir grafikos vartojimas kompiuterių tinkluose ir internete tampa priimtinesnis. Trumpai tariant, didelio našumo suspaudimo technologijos davė pagrindą atsirasti naujoms aplikacijoms — skaitmeninė biblioteka, skaitmeniniai archyvai, videokonferencija (bendravimas televizijos pagalba), telemedicina, skaitmeninės pramogos ir pan. Yra ir daugiau duomenų suspaudimo pranašumų (pavyzdžiui, duomenų bazių prieigai). Duomenų suspaudimas gali padidinti duomenų bazės našumą todėl, kad daugiau suspaustų įrašų gali būti supakuota duotoje buferio dalyje.

Nors duomenų suspaudimas pateikia daug privalumų, tačiau negalime nutylėti ir kai kurių esminių trūkumų (įvairūs pašaliniai efektai, prarandama svarbi informacija ir pan.), priklausančių nuo aplikacijų panaudojimo srities ir duomenų jautrumo.

Šiame darbe realizuojami skaitmeninių vaizdų kodavimui pasirinkti metodai sukuriant jiems programinę įrangą, kurios pagalba sprendžiamos iškilusios vaizdų kodavimo problemos. Pagrindiniai darbo akcentai:

- ✓ Diskrečiosios Haaro transformacijos ir progresyvaus vaizdų kodavimo algoritmo EZW (Embedded Zerotree Wavelet) „suderinamumas“;
- ✓ Pristatoma nauja lokaliai progresyvaus skaitmeninių vaizdų kodavimo versija.

2. SKAITMENINIŲ VAIZDŲ SUGLAUDINIMO SPEKTRINĖJE SRITYJE METODŲ APŽVALGA

Diskrečiosios transformacijos yra gana patikimas, efektyvus bei galingas įrankis, naudojamas skaitmeninių vaizdų ir signalų apdorojimo srityje. Trumpai aptarsime svarbiausias (praktiniu požiūriu) diskrečiąsias transformacijas, jų savybes bei greito skaičiavimo algoritmus.

Tegul $[X(m)] = X = (X(0)\ X(1)\ \ldots\ X(N-1))^T$ yra vienmatis skaitmeninis vaizdas (duomenų vektorius); čia: $m = 0, 1, \ldots, N-1$; $N = 2^n$, $n \in \mathbf{N}$. Bendru atveju, diskrečioji transformacija (DT) šiam duomenų vektoriui apibrėžiama taip:

$$[Y(k)] = Y = (Y(0)\ Y(1)\ \ldots\ Y(N-1))^T = \frac{1}{N}\ T \cdot X; \tag{2.1}$$

čia: Y – diskretusis vaizdo X spektras; T – DT matrica; pagrindinė matricos savybė – jos ortogonalumas, t.y., jeigu matricos elementus pažymėsime t_{ij} ($i, j = 0, 1, \ldots, N-1$), tai ortogonalumo sąlygą galėsime užrašyti taip:

$$\sum_{j=0}^{N-1} t_{kj} \cdot \bar{t}_{lj} = \begin{cases} 0, kai\ k \neq l, \\ N, kai\ k = l; \end{cases} \tag{2.2}$$

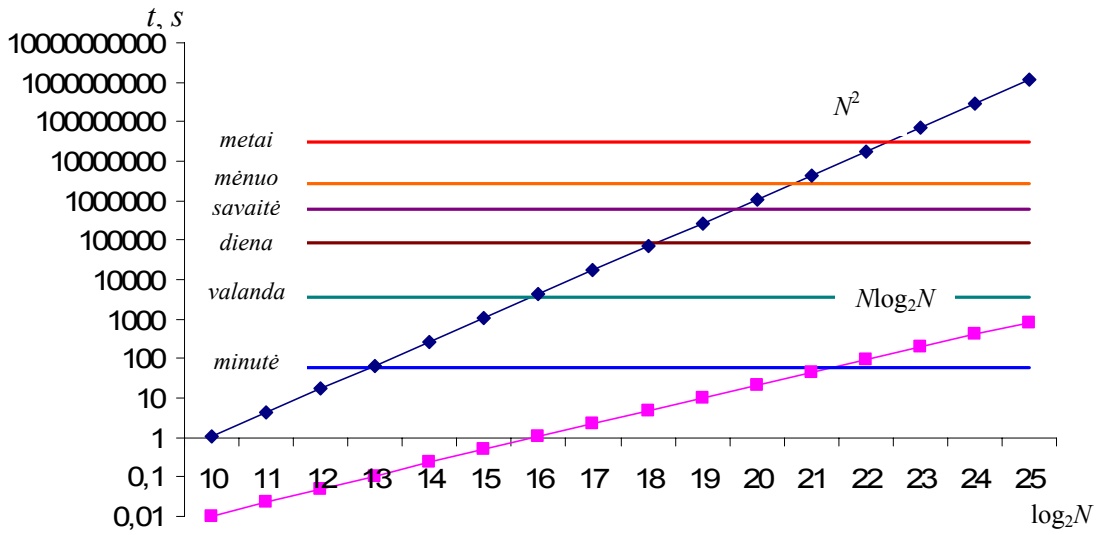
$k, l \in \{0, 1, \ldots, N-1\}$; $\bar{t}_{lj}$ – kompleksinis jungtinis skaičius.

Dažnai DT matricos T eilutės vadinamos DT baziniais vektoriais. Iš matricos T ortogonalumo sąlygos išplaukia, jog $\overline{T^T}$ ir T^{-1} yra proporcingos matricos, t.y. $\overline{T^T} = N \cdot T^{-1}$. Tai leidžia apibrėžti atvirkštinę diskrečiąją transformaciją (ADT)

$$X = [X(m)] = \overline{T^T} \cdot Y. \tag{2.3}$$

Išraiškos (2.1) ir (2.3) sudaro transformacijų porą, o tai reiškia, kad pereinant nuo duomenų vektoriaus (vaizdo) prie spektro, informacija neprarandama. Šis faktas, taipogi, leidžia teigti, jog DT ir ATD skaičiavimo algoritmai turėtų būti panašūs.

Atkreiptinas dėmesys į DT (tuo pačiu ADT) algoritmų (skaičiavimo) sudėtingumą. Norint apskaičiuoti duomenų vektoriaus $[X(m)]$ spektrą, reikia N^2 daugybos operacijų. Tai daug ar mažai? Praktikoje tenka dirbti su vaizdais, kai $N = 2^{18} \div 2^{22}$. Tarus, jog skaičiavimo įrankis (kompiuteris, kita ESM) gali atlikti $\sim$ 1 mln. operacijų (daugybos) per sekundę, gautume tokią diagramą (2.1 pav.):



2.1 pav. DT spektro radimui reikalingos laiko sąnaudos (N^2 – pagal apibrėžimą; $N\log_2 N$ – greito DT skaičiavimo algoritmas)

Panaši situacija ir dėl ADT. Taigi, praktinis DT panaudojimas, kai N yra pakankamai didelis skaičius, tampa problematišku. Padėtį gelbsti greiti DT skaičiavimo algoritmai, leidžiantys sumažinti daugybos operacijų skaičių nuo N^2 iki $N\log_2 N$ (2.1 pav.). Apie juos užsiminsime tolimesniuose skyreliuose.

2.1 Diskrečiosios transformacijos ir jų greito skaičiavimo algoritmai

2.1.1 Diskrečioji Furjė transformacija (DFT)

Imkime vienmatį skaitmeninį vaizdą (duomenų vektorių) $[X(m)] = X = (X(0), X(1),..., X(N-1))^T$; čia $m = 0,1,..., N-1$, $N = 2^n$, $n \in \mathbf{N}$. Tiesioginė DFT šiam vaizdui apibrėžiama taip:

$$C_X(k) = \frac{1}{N}\sum_{m=0}^{N-1} X(m)\cdot W^{km} = \frac{1}{N}[X(0) + X(1)\cdot W^k + X(2)\cdot W^{2k} + ... + X(N-1)\cdot W^{(N-1)k}]; \quad \textbf{(2.4)}$$

čia: $[C_X(k)] = C_X = (C_X(0)\ C_X(1)\ ...\ C_X(N-1))^T$ yra DFT koeficientų vektorius (diskretusis DFT spektras) vaizdui X; $k = 0, 1, \ldots, N-1$; $W = e^{-i\frac{2\pi}{N}}$, $i^2 = -1$.

Atvirkštinė DFT nusakoma lygybe:

$$X(m) = \sum_{k=0}^{N-1} C_X(k)\cdot \overline{W^{km}} = \sum_{k=0}^{N-1} C_X(k)\cdot W^{-km}; \quad \textbf{(2.5)}$$

$m = 0, 1, \ldots, N-1$.

Įvedę DFT matricą T, t.y.

$$T = [W^{km}] = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & W & W^2 & \dots & W^{N-1} \\ 1 & W^2 & W^4 & \dots & W^{2(N-1)} \\ \dots & \dots & \dots & \dots & \dots \\ 1 & W^{N-1} & W^{(N-1)2} & \dots & W^{(N-1)(N-1)} \end{pmatrix} \begin{matrix} k=0 \\ k=1 \\ k=2 \\ \dots \\ k=N-1 \end{matrix},$$
$$m=0 \quad m=1 \quad m=2 \quad \dots \quad m=N-1$$

(2.4) ir (2.5) išraiškas galime pateikti matriciniu pavidalu:

$$C_X = \frac{1}{N} T \cdot X, \tag{2.6}$$

$$X = \overline{T^T} \cdot C_X = \overline{T} \cdot C_X. \tag{2.7}$$

Suformuluosime kelias pagrindines DFT savybes:

- ✓ Tiesiškumas. Jeigu $X(m) \leftrightarrow C_X(k)$, $Y(m) \leftrightarrow C_Y(k)$ ir $Z(m) = \alpha X(m) + \beta Y(m)$ (su visais $m = 0,1,...,N-1$; $\alpha, \beta \in \mathbf{R}$), tai $Z(m) \leftrightarrow C_Z(k) = \alpha C_X(k) + \beta C_Y(k)$.
- ✓ Sąsuka. Jeigu X ir Y yra realieji duomenų vektoriai (interpretuojami kaip N–periodinės sekos), tai jų sąsūka (sąsūkos vektorius $[Z(m)]$) apibrėžiama lygybe

$$Z(m) = \frac{1}{N} \sum_{h=0}^{N-1} X(h) \cdot Y(m-h) \ \ (m,\ h = 0,\ 1,\ \dots,\ N-1). \tag{2.8}$$

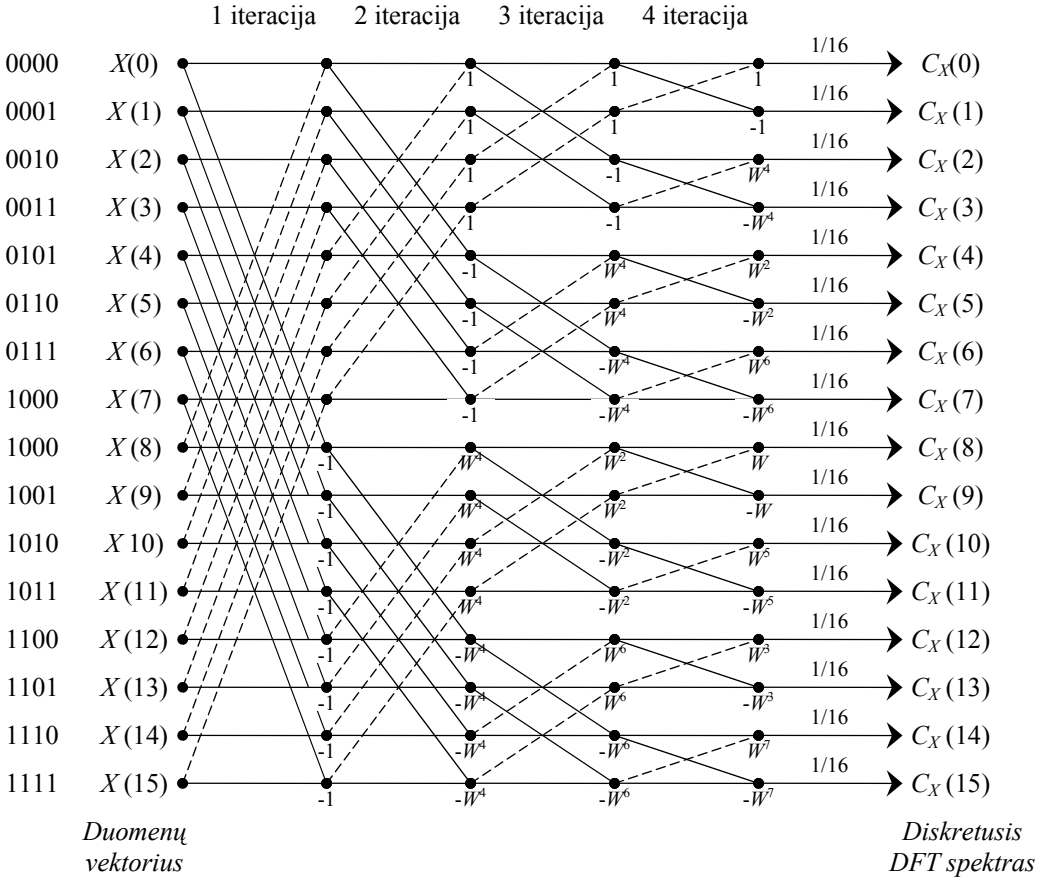
  Be to, jeigu $X(m) \leftrightarrow C_X(k)$, $Y(m) \leftrightarrow C_Y(k)$, tai $Z(m) \leftrightarrow C_Z(k) = C_X(k) \cdot C_Y(k)$.
- ✓ Koreliacija. Jeigu X ir Y yra realieji duomenų vektoriai, tai jų koreliacija apibrėžiama lygybe

$$\hat{Z}(m) = \frac{1}{N} \sum_{h=0}^{N-1} X(h) \cdot Y(m+h) \ \ (m,\ h = 0,\ 1,\ \dots,\ N-1). \tag{2.9}$$

  Be to, $\hat{Z}(m) \leftrightarrow C_{\hat{Z}}(k) = \overline{C_X(k)} \cdot C_Y(k)$.

Kaip jau buvo minėta šio skyrelio pradžioje, praktinis šios transformacijos panaudojimas yra ribotas. Viena iš priežasčių – laiko sąnaudos (2.1 pav.), siejamos su DFT spektro apskaičiavimu, kai N yra didelis skaičius ($N \geq 2^{20}$). Situacija kardinaliai pasikeitė, kai amerikiečių matematikai I.W.Cooley ir I.W.Tukey pirmąkart (1965 m.) aprašė greitą DFT skaičiavimo

procedūrą – greitą Furjė transformaciją (GFT algoritmą). GFT algoritmo veikimą patogu demonstruoti, panaudojant taip vadinamą „signalinį“ grafą (2.2 pav.).



2.2 pav. GFT „signalinis“ grafas, kai $N = 16$

Pastebėsime, jog GFT „dirba“ tik tuo atveju, kai $N = 2^n$ ($n \in \mathbf{N}$), DFT spektro radimui reikia atlikti $n = \log_2 N$ iteracijų. Pirmojoje iteracijoje dalyvauja visi duomenų vektoriaus (vaizdo) elementai, antrojoje iteracijoje – formuojamos ir apdorojamos dvi atskiros grupės, trečiojoje – keturios atskiros grupės, pagaliau ketvirtojoje iteracijoje dalyvauja jau aštuonios grupės (po du duomenų vaizdo elementus. Svoriai W ($W = e^{-i\frac{2\pi}{N}} = e^{-i\frac{2\pi}{16}} = \cos\frac{\pi}{8} - i\sin\frac{\pi}{8}$) parenkami, kaip parodyta 2.1 lentelėje.

Būtina pabrėžti, jog DFT spektro (vektoriaus C_X) koeficientai gaunami „inversine“ tvarka. Reikalinga atlikti papildomą dvejetainę inversiją. GFT algoritmo privalumas yra tas, kad DFT spektro skaičiavimas nereikalauja papildomos atminties – spektras formuojamas duomenų vektoriaus vietoje. Tačiau GFT turi ir trūkumą – jeigu $X = (X(0)\ X(1)\ \ldots\ X(N-1))^T$ yra realusis duomenų vektorius, jo DFT spektras yra kompleksinis.

2.1 lentelė. Svorių *W* nustatymo lentelė

Dešimtainė forma	Dvejetainė forma	Inversinė dvejetainė forma	Dešimtainė forma	1 iter.	2 iter.	3 iter.	4 iter.
0	000	000	0	$\pm W^0$	$\pm W^0$	$\pm W^0$	$\pm W^0$
1	001	100	4		$\pm W^4$	$\pm W^4$	$\pm W^4$
2	010	010	2			$\pm W^2$	$\pm W^2$
3	011	110	6			$\pm W^6$	$\pm W^6$
4	100	001	1				$\pm W$
5	101	101	5				$\pm W^5$
6	110	011	3				$\pm W^3$
7	111	111	7				$\pm W^7$

Diskrečiąją Furjė transformaciją galima nesunkiai apibendrinti dvimačiams bei trimačiams vaizdams.

Tarkime, kad $[X(m_1, m_2)]$ yra dvimatis skaitmeninis vaizdas (dvimatis duomenų masyvas):

$$[X(m_1, m_2)] = \begin{pmatrix} X(0,0) & X(0,1) & ... & X(0, N_2 - 1) \\ X(1,0) & X(1,1) & ... & X(1, N_2 - 1) \\ ... & ... & ... & ... \\ X(N_1 - 1,0) & X(N_1 - 1,1) & ... & X(N_1 - 1, N_2 - 1) \end{pmatrix}. \quad \textbf{(2.10)}$$

čia: $m_1 = 0, 1, \ldots, N_1 - 1$; $m_2 = 0, 1, \ldots, N_2 - 1$; $N_1 = 2^{n_1}$; $N_2 = 2^{n_2}$; $n_1, n_2 \in \mathbf{N}$.

Dvimatė DFT skaitmeniniam vaizdui (duomenų masyvui) $[X(m_1, m_2)]$ apibrėžiama lygybe:

$$C_{XX}(k_1, k_2) = \frac{1}{N_1 \cdot N_2} \sum_{m_1=0}^{N_1-1} \sum_{m_2=0}^{N_2-1} X(m_1, m_2) \cdot W_1^{k_1 \cdot m_1} \cdot W_2^{k_2 \cdot m_2}; \quad \textbf{(2.11)}$$

čia: $k_1 = 0, 1, \ldots, N_1 - 1$; $k_2 = 0, 1, \ldots, N_2 - 1$. Iš (2.10) išraiškos tiesiogiai išplaukia, jog dvimačio DFT spektro koeficientai gaunami skaičiuojant vienmatę DFT kiekvienam vaizdo $[X(m_1, m_2)]$ stulpeliui, o po to kiekvienai gauto tarpinio masyvo eilutei. Taip gaunama aibė iš $N_1 \cdot N_2$ spektrinių koeficientų.

Atvirkštinės dvimatės DFT išraiška tokia:

$$X(m_1, m_2) = \sum_{k_2=0}^{N_2-1} \sum_{k_1=0}^{N_1-1} C_{XX}(k_1, k_2) \cdot W_1^{-k_1 \cdot m_1} \cdot W_2^{-k_2 \cdot m_2}; \quad \textbf{(2.12)}$$

čia: $m_1 = 0, 1, \ldots, N_1 - 1$; $m_2 = 0, 1, \ldots, N_2 - 1$.

Kalbant apie trimatę DFT (tiek tiesioginę, tiek atvirkštinę), galima teigti, jog ji apibrėžiama ir realizuojama remiantis visiškai analogiškais samprotavimais, kaip ir dvimačiu atveju, būtent:

$$C_{XXX}(k_1,k_2,k_3)=\frac{1}{N_1\cdot N_2\cdot N_3}\sum_{m_3=0}^{N_3-1}\sum_{m_2=0}^{N_2-1}\sum_{m_1=0}^{N_1-1}X(m_1,m_2,m_3)\cdot W_1^{k_1\cdot m_1}\cdot W_2^{k_2\cdot m_2}\cdot W_3^{k_3\cdot m_3}\ ; \quad \textbf{(2.13)}$$

$$X(m_1,m_2,m_3)=\sum_{k_3=0}^{N_3-1}\sum_{k_2=0}^{N_2-1}\sum_{k_1=0}^{N_1-1}C_{XXX}(k_1,k_2,k_3)\cdot W_1^{-k_1\cdot m_1}\cdot W_2^{-k_2\cdot m_2}\cdot W_3^{-k_3\cdot m_3}\ ; \quad \textbf{(2.14)}$$

čia: k_1, $m_1 = 0, 1, \ldots, N_1 - 1$; k_2, $m_2 = 0, 1, \ldots, N_2 - 1$; k_3, $m_3 = 0, 1, \ldots, N_3 - 1$. Faktiškai, trimatės DFT atveju būtina $(N_1N_2 + N_1N_3 + N_2N_3)$ kartų taikyti vienmatę DFT (pagal kiekvieną koordinatę atskirai).

Pagrindinius DFT praktinio panaudojimo sritys – spektrinė signalų analizė, koreliacinė signalų analizė, skaitmeninių signalų ir vaizdų apdorojimas ir kt.

2.1.2 Volšo ir Adamaro diskrečioji transformacija (VAT)

Pilną ortonormuotą Volšo funkcijų sistemą $\{wal(i, t)\}$, $t \in [0, 1)$, įvedė amerikiečių matematikas I.L.Walsh. Pati Volšo funkcijų sistema paprastai pateikiama kuriuo nors vienu iš trijų variantų, besiskiriančių atskirų funkcijų išdėstymo tvarka sistemoje, būtent:

(1) Išdėstymas pagal Volšą — $\{wal_w(i, t)\}$;

(2) Išdėstymas pagal Adamarą — $\{wal_h(k, t)\}$;

(3) Išdėstymas pagal Pėlį — $\{wal_p(j, t)\}$.

Diskretizavus šias funkcijas (bet kurį variantą) ir paėmus baigtinį jų skaičių, apibrėžiamos ir konstruojamos atitinkamos diskrečiosios transformacijos — VAT_w, VAT_h, VAT_p. Visos jos vadinamos Volšo ir Adamaro diskrečiosiomis transformacijomis.

Patogiausia apibrėžti ir komentuoti VAT_h, kadangi šios diskrečiosios transformacijos matrica T gaunama, panaudojant rekurentinius sąryšius:

$$T=H_h(n)=\begin{pmatrix}H_h(n-1) & H_h(n-1)\\ H_h(n-1) & -H_h(n-1)\end{pmatrix},\ H_h(1)=\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix},\ n=2, 3, \ldots \quad \textbf{(2.15)}$$

Volšo ir Adamaro diskrečioji transformacija VAT_h (išdėstymas pagal Adamarą) vienmačiam vaizdui $X = [X(m)] = (X(0)\ X(1)\ \ldots\ X(N-1))^T$ apibrėžiama lygybe:

$$[S_X(k)] = S_X = (S_X(0)\ S_X(1)\ \ldots\ S_X(N-1))^T = \frac{1}{N}H_h(n)\cdot X. \quad \textbf{(2.16)}$$

Kadangi $H_h^T(n) \cdot H_h(n) = N \cdot E$, tai atvirkštinė VAT$_h$ nusakoma lygybe $X = H_h(n) \cdot S_X$.

Kitų dviejų Volšo ir Adamaro transformacijų (VAT$_w$ ir VAT$_p$) atskirai čia neapibrėšime.

Volšo ir Adamaro transformacijos, kokie jų baziniai vektoriai bebūtų, pasižymi savybėmis, kurių dėka jos plačiai naudojamos praktikoje, būtent:

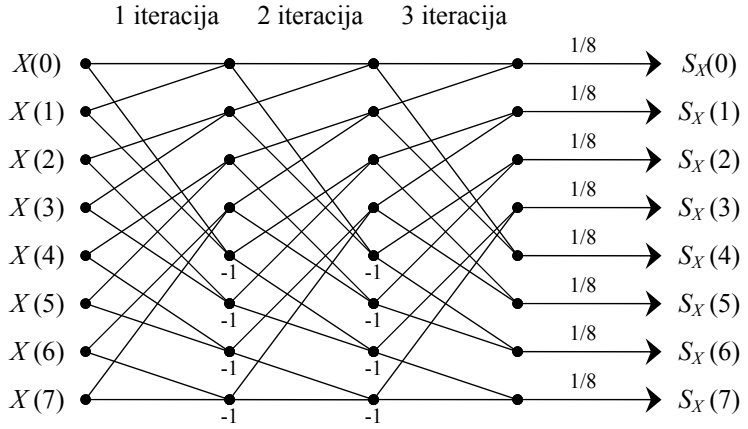
(1) Jų matricų elementai įgyja tik dvi reikšmės — 0 arba 1. Tai labai patogi kompiuterinė realizacija; gana paprasti greito skaičiavimo algoritmai (2.3 pav.).

(2) Matrica yra simetriška, t.y. $H_h(k) = H_h^T(k)$.

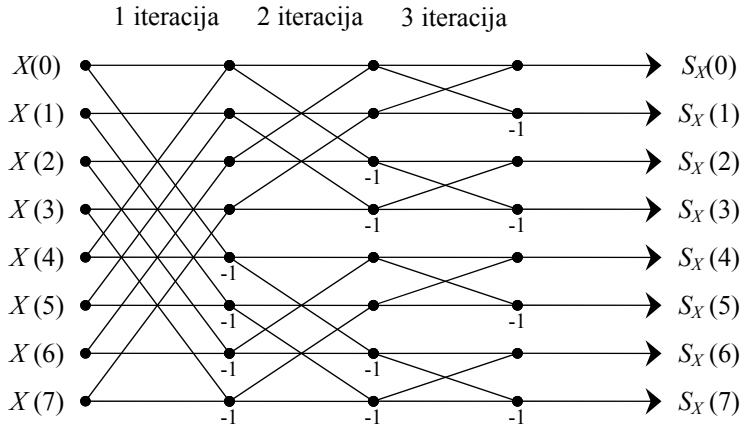
(3) Matrica yra ortogonali, t.y. $H_h^T(k) \cdot H_h(k) = 2^k I_k$, čia I_k – vienetinė ($2^k \times 2^k$) matrica.

(4) Adamaro matrica ir jai atvirkštinė yra proporcingos viena kitai, t.y.

$$H_h^{-1}(k) = \frac{1}{2^k} H_h(k).$$



(a)



(b)

2.3 pav. VAT$_h$ skaičiavimo algoritmų „signaliniai“ grafai: (a) „rankinis“; (b) „Cooley-Tukey“ tipo

Panašiai kaip ir DFT, ši transformacija plačiai naudojama signalų ir vaizdų apdorojime (dispersiniai kriterijai, hiperboliniai filtrai ir pan.).

2.1.3 Diskrečioji kosinusinė transformacija (DKT)

Nagrinėkime vienmatį skaitmeninį vaizdą (duomenų vektorių) $X = [X(m)] = (X(0), X(1),..., X(N-1))^T$; čia $N = 2^n$; $n \in \mathbf{N}$. Diskrečiąją kosinusinę transformaciją (DKT) vaizdui $[X(m)]$ apibūdina lygybė:

$$L_X(0) = \frac{1}{\sqrt{N}} \sum_{m=0}^{N-1} X(m),\ L_X(k) = \sqrt{\frac{2}{N}} \sum_{m=0}^{N-1} X(m) \cdot \cos\frac{(2m+1)k}{2N}; \qquad \textbf{(2.17)}$$

čia: $k = 0, 1, \ldots, N-1$; $L_X = [L_X(k)] = (L_X(0)\ L_X(1)\ \ldots\ L_X(N-1))^T$ – DKT spektras.

Transformaciją DKT nesunku užrašyti matricine išraiška: $L_X = K(n) \cdot X$, kur transformacijos matrica

$$K(n) = \begin{pmatrix} \frac{1}{\sqrt{N}} & \frac{1}{\sqrt{N}} & \ldots & \frac{1}{\sqrt{N}} \\ \sqrt{\frac{2}{N}}\cos\frac{\pi}{2N} & \sqrt{\frac{2}{N}}\cos\frac{3\pi}{2N} & \ldots & \sqrt{\frac{2}{N}}\cos\frac{(2N-1)\pi}{2N} \\ \ldots & \ldots & \ldots & \ldots \\ \sqrt{\frac{2}{N}}\cos\frac{(N-1)\pi}{2N} & \sqrt{\frac{2}{N}}\cos\frac{3(N-1)\pi}{2N} & \ldots & \sqrt{\frac{2}{N}}\cos\frac{(2N-1)(N-1)\pi}{2N} \end{pmatrix}. \qquad \textbf{(2.18)}$$

Kadangi $K^T(n) \cdot K(n) = E$ (E – vienetinė matrica), tai atvirkštinė diskrečioji kosinusinė transformacija (ADKT) nusakoma lygybe:

$$X(m) = \frac{1}{\sqrt{N}} L_X(0) + \sqrt{\frac{2}{N}} \sum_{k=1}^{N-1} L_X(k) \cdot \cos\frac{(2m+1)\pi}{2N} \text{ arba } X = K^T(n) \cdot L_X; \qquad \textbf{(2.19)}$$

$m = 0, 1, \ldots, N-1$.

Dvimatės DKT ir ADKT yra tokios:

$$\begin{aligned} [L_{XX}(k_1, k_2)] &= K(n_1) \cdot [X(m_1, m_2)] \cdot K^T(n_2), \\ [X(m_1, m_2)] &= K^T(n_1) \cdot [L_{XX}(k_1, k_2)] \cdot K(n_2). \end{aligned} \qquad \textbf{(2.20)}$$

Viena svarbesnių DKT savybių, dėl kurios ji plačiai panaudojama praktikoje, yra tai, jog ši transformacija labai gerai „pakuoja“ vaizde sukauptą informaciją, t.y. pervedant vaizdą į

spektrinę sritį, informacija kaupiama žemuose dažniuose. Būtent dėl šios „pakavimo“ savybės DKT sėkmingai panaudota dvimačių vaizdų suglaudinimo standarte JPEG.

Ypatingas vaidmuo skaitmeniniame signalų ir vaizdų apdorojime tenka diskrečiosioms bangelių transformacijoms (DBT). Bene paprasčiausiam DBT šeimos atstovui – diskrečiajai Haaro transformacijai – šiame darbe skiriamas pagrindinis dėmesys.

2.1.4 Diskrečioji Haaro transformacija

Ortogonali Haaro funkcijų sistema (bangelės) paprastai pateikiama tokia išraiška:

$$haar(0,0,x) = 1, \quad x \in [0,1),$$

$$haar(r,m,x) = \begin{cases} \sqrt{2^r}, (m-1)/2^r \le x < (m-1/2)/2^r, \\ -\sqrt{2^r}, (m-1/2)/2^r \le x < m/2^r, \\ 0, kitu \quad atveju; \end{cases} \tag{2.21}$$

čia: r = 0, 1, 2, ...; m = 1, 2, ..., 2^r, [8], [11]. Haaro (bangelių) skleidiniai turi daug įdomių savybių, kurios nėra būdingos kitokio tipo skleidiniams (Furjė, Volšo ir pan.). Viena jų yra lokalizavimo erdvėje savybė — jei domina skleidžiamos funkcijos (signalo) „elgesys“ intervale $[a;b] \subset [0,1)$, pakanka analizuoti tiktai nedidelį Haaro skleidinio koeficientų poaibį. Atkreiptinas dėmesys ir į tai, kad Haaro bangelės yra sukurtos iš vienos („motininės“) bangelės *haar*(0, 1, x), $x \in [0, 1)$, naudojant diadinius mastelio keitinius ir sveikaskaitinius postūmius. Neabejotinai, lokalizavimas erdvėje ir pastaroji savybė yra Haaro (bangelių) skleidinių skiriamasis bruožas. Dėka šių savybių Haaro bangelės (Haaro skleidiniai, diskretusis Haaro spektras) sėkmingai panaudojamos skaitmeninių vaizdų (signalų) apdorojimo sferoje.

Imkime duomenų vektorių (vienmatį skaitmeninį vaizdą): $X = [X(m)] = (X(0), X(1),..., X(N-1))^T$; čia $N = 2^n$; $n \in \mathbf{N}$.

Diskrečioji Haaro transformacija (HT) vaizdui X apibrėžiama lygybe:

$$Y = \frac{1}{N} H(n) \cdot X; \tag{2.22}$$

čia: $H(n)$ – ortogonali n–os eilės Haaro matrica ($N \times N$); $Y = [Y(k)] = (Y(0)\ Y(1)\ \ldots\ Y(N-1))^T$ – diskretusis Haaro spektras vaizdui X. Matrica $H(n)$ gaunama, diskretizuojant baigtinį skaičių pirmųjų Haaro funkcijų iš aibės $\{haar(r, m, x)\}$, $0 \le x < 1$. Pavyzdžiui, kai $N = 8 = 2^3$, HT matrica yra tokia:

$$H(3)=\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{pmatrix}, \qquad \textbf{(2.23)}$$

Kadangi matrica $H(n)$ yra ortogonali, t.y. $H(n)\cdot H^T(n)=N\cdot E(n)$, tai atvirkštinė Haaro transformacija (AHT) nusakoma lygybe

$$X=H^T(n)\cdot Y\,. \qquad \textbf{(2.24)}$$

Žinoma keletas greitų HT skaičiavimo algoritmų (Cooley-Tukey tipo, Endrew‘s ir kt.). Endrew‘s algoritmų signaliniai grafai (tiesioginei ir atvirkštinei HT apskaičiuoti) yra pateikti 2.4–ame paveikslėlyje. Tai plačiai praktikoje taikomi tipiniai greiti HT ir AHT skaičiavimo algoritmai.

Imkime dvimatį skaitmeninį vaizdą

$$X=[X(m_1,m_2)]=\begin{pmatrix} X(0,0) & X(0,1) & ... & X(0,N_2-1) \\ X(1,0) & X(1,1) & ... & X(1,N_2-1) \\ ... & ... & ... & ... \\ X(N_1-1,0) & X(N_1-1,1) & ... & X(N_1-1,N_2-1) \end{pmatrix}; \qquad \textbf{(2.25)}$$

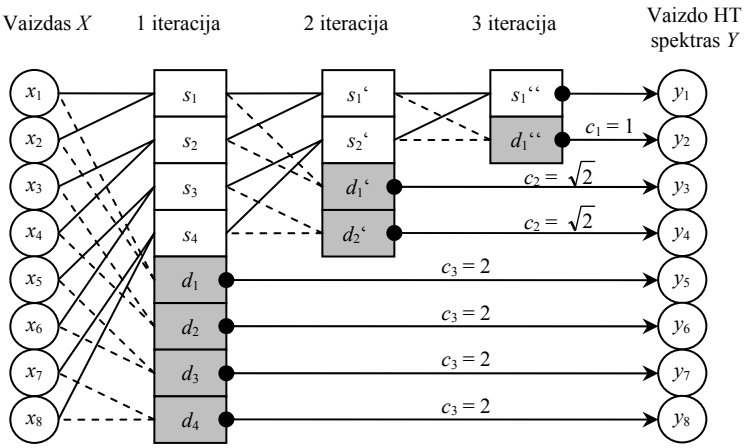
čia: $m_1=0,1,...,N_1-1$; $m_2=0,1,...,N_2-1$; $N_1=2^{n_1}$; $N_2=2^{n_2}$; $n_1,n_2\in\mathbf{N}$. Vaizdo $[X(m_1, m_2)]$ diskrečioji Haaro transformacija apibrėžiama lygybe:

$$[Y(k_1,k_2)]=\frac{1}{N_1\cdot N_2}\,H(n_1)\cdot[X(m_1,m_2)]\cdot H^T(n_2), \qquad \textbf{(2.26)}$$

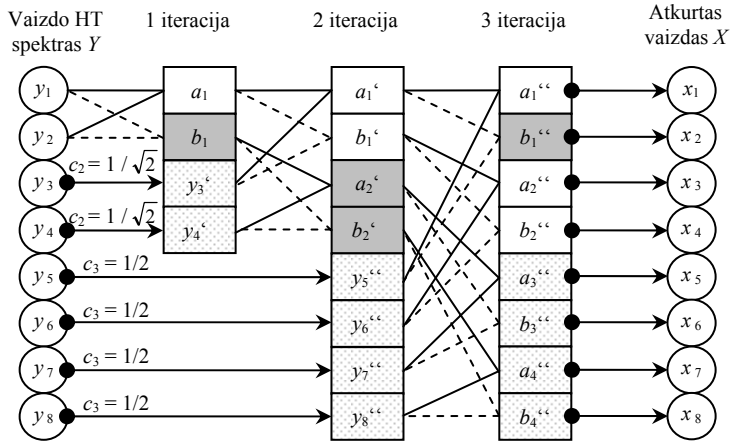
kur $H(n_1)$ ir $H(n_2)$ yra atitinkamai n_1–os ir n_2–os eilių Haaro transformacijos matricos.

Pastebėsime, jog dvimatė atvirkštinė Haaro transformacija gali būti apskaičiuojama, taikant formulę:

$$[X(m_1,m_2)]=H^T(n_1)\cdot[Y(k_1,k_2)]\cdot H(n_2). \qquad \textbf{(2.27)}$$



(a)



(b)

2.4 pav. Greitų HT ir AHT transformacijų apskaičiavimo algoritmų „signaliniai“ grafai: (a) HT; (b) AHT ($N = 2^3 = 8$; $c_r = c_{r-1} \cdot \sqrt{2}$, $r = 2, 3$; $c_1 = 1$)

Kaip ir ankstesnių diskrečiųjų transformacijų (DFK, DKT ir pan.) atveju, HT dvimačiams vaizdams apskaičiuojama, panaudojant vienmatę HT, būtent:

(1) Kiekvienam dvimačio vaizdo $[X(m_1, m_2)]$ stulpeliui $(X(0, m_2)\ X(1, m_2)\ ...\ X(N_1 - 1, m_2))^T$ $(m_2 \in \{0, 1, \ldots, N_2 - 1\})$ atliekama vienmatė diskrečioji HT. Gaunamas tarpinis masyvas $[X_1(m_1, k_2)]$.

(2) Kiekvienai tarpinio masyvo $[X_1(m_1, k_2)]$ eilutei $(X_1(m_1, 0)\ X_1(m_1, 1)\ ...\ X_1(m_1, N_2 - 1))$ $(m_1 \in \{0, 1, \ldots, N_1 - 1\})$ vėlgi atliekama diskrečioji HT. Gaunamas dvimačio vaizdo $[X(m_1, m_2)]$ Haaro spektras $[Y(k_1, k_2)]$.

Spektriniai Haaro koeficientai (tiek vienmačiu, tiek dvimačiu atvejais) pasižymi labai įdomiomis savybėmis, kurių neturi kitų diskrečiųjų transformacijų (Furjė, Volšo ir pan.)

spektriniai koeficientai. Tai lokalizavimo erdvėje savybė, sąryšiu su atskirais vaizdo fragmentais buvimas ir pan. Šias savybes detaliau aptarsime 2.4.1 ir 2.4.2 skyreliuose.

2.2 Vaizdų suglaudinimo spektrinėje srityje metodų apžvalga

Šiame skyrelyje trumpai pristatysime ir pakomentuosime kelis skaitmeninius vaizdų suglaudinimo spektrinėje srityje metodus (požiūrius). Tai klasikinis vaizdų suglaudinimo metodas, taikantis dispersinius kriterijus, JPEG standartas bei hiperboliniai vaizdų filtrai.

2.2.1 Dispersiniai kriterijai

Imkime vienmatį diskretųjį vaizdą (duomenų vektorių) $X = (X(0)\ X(1)\ \ldots\ X(N-1))^T$, priklausantį tam tikram N–mačių vektorių klasei S, $N = 2^n$, $n \in \mathbf{N}$.

Tarkime, kad $Y = (Y(0)\ Y(1)\ \ldots\ Y(N-1))^T = \frac{1}{N} T \cdot X$ (T – ortonormuota matrica) yra diskrečiosios transformacijos spektras vektoriui X.

Duomenų suspaudimo uždavinys yra formuluojamas taip – panaudojant tam tikrus kriterijus, atrinkti transformacijos koeficientų poaibį $Y(K)$. Likusieji $(N-K)$ spektro Y elementai atmetami. Ši procedūra neturi iššaukti kiek žymesnės paklaidos, atkuriant vektorių X, t.y. gaunant jo įvertį $\widetilde{X}$.

Sprendžiant šį uždavinį, ieškoma tokios diskrečiosios transformacijos, kuri, iš vienos pusės, vienareikšmiškai vaizduotų vektorių X, ir, iš kitos pusės, būtų optimali (esant fiksuotam duomenų suspaudimo laipsniui N / K) vidutinės kvadratinės paklaidos kriterijaus δ atžvilgiu (2.28).

$$\delta = \delta(K) = \frac{1}{N} \sum_{m=0}^{N-1} (\widetilde{X}(m) - X(m))^2 \ . \qquad \textbf{(2.28)}$$

Įrodyta, kad jeigu Σ_x yra signalų klasės S kovariacinė matrica, tai optimalios diskrečiosios transformacijos baziniai vektoriai turi sutapti su matricos Σ_x tikriniais vektoriais. Tada kovariacinė matrica (spektrinėje srityje) Σ_y turi išraišką

$$\Sigma_y = T \cdot \Sigma_x \cdot T^T = \mathrm{diag}(\lambda_0, \lambda_1, \ldots, \lambda_{N-1}), \qquad \textbf{(2.29)}$$

kur $i = 0, 1, ..., N-1$, λ_i yra tikrinės reikšmės, atitinkančios matricos Σ_y tikrinius vektorius. Minimali vidutinė kvadratinė paklaida $\delta_{\min}(K)$ šiuo atveju lygi

$$\delta_{\min}(K) = \sum_{S=K}^{N-1} \lambda_{i_s} . \quad \textbf{(2.30)}$$

Akivaizdu, kad transformacijos koeficiento $Y(k)$ panaudojimo, vaizduojant duomenų vektorių X, efektyvumas nusakomas jam atitinkančia tikrine reikšme λ_k. Jeigu koeficientas $Y(k)$ atmetamas, t.y. neįjungiamas į saugojimui skirtą poaibį $Y(k)$, tai vidutinė kvadratinė paklaida padidėja atitinkama tikrine reikšme λ_k. Tokiu būdu, į poaibį $Y(K)$ atrenkami tik tie K transformacijos koeficientai, kuriems atitinka didžiausios matricos Σ_x tikrinės reikšmės (arba didžiausi matricos Σ_y įstrižainės elementai (dispersijos!)), o likusieji $(N - K)$ elementai atmetami, kadangi juos galima ir tikslinga vėliau (atkuriant vektorių X) pakeisti iš anksto apibrėžtomis (signalų klasei S) konstantomis c_{i_s}, $s = K, K+1, ..., N-1$.

Aprašytas saugojimui skirtų transformacijos koeficientų išrinkimo kriterijus yra vadinamas dispersiniu kriterijumi. Visiškai panaši dispersinio kriterijaus interpretacija daugiamačiu atveju, kurio čia detaliau neaptarinėsime.

Vienas pagrindinių klasikinio duomenų suspaudimo metodo trūkumų yra tas, kad apdorojant signalą (tiek vienmačiu, tiek daugiamačiu atvejais), reikia žinoti, kokiai diskretizuotų signalų klasei jis priklauso, t.y. būtina žinoti tų klasių kovariacines matricas. Tai pateisinama, kai minėtu būdu apdorojamas pakankamai didelis skaičius vaizdų, priklausančių vienai ir tai pačiai klasei S. Antra, tikrinių vektorių skaičiavimas, kai N yra didelis skaičius, gana komplikuotas. Ir trečia, atlikti eksperimentai rodo, jog realus maksimalus informacijos suspaudimo laipsnis N / K neviršija 3 ÷ 5, o šito, ypač daugiamačiu atveju, nepakanka.

2.2.2 JPEG standartas

JPEG – informaciją prarandantis vaizdų kodavimo metodas, skirtas spalvotiems ir nespalvotiems vaizdams bei fotografijoms koduoti (glaudinti). Metodas pavadintas pagal šį standartą kūrusios jungtinės ekspertų grupės pavadinimą (Joint Photographic Experts Group) [5]. Šis standartas aprašo du suglaudinimo metodus: informaciją prarandantį ir informaciją išsaugantį (neprarandantį). Pirmasis metodas paremtas DKT taikymu, kai tuo tarpu antrasis metodas naudoja aritmetinį (arba Huffmano) kodavimą. DKT paremto metodo įgyvendinimas yra privalomas visoms JPEG kodavimo (dekodavimo) realizacijoms, todėl šį metodą panagrinėsime šiek tiek detaliau (privalomas ir dažniausiai naudojamas šio standarto priemones).

Informaciją prarandantis JPEG suglaudinimas nespalvotiems vaizdams realizuojamas tokia tvarka:

(1) Pradinis skaitmeninis vaizdas suskaidomas blokeliais 8×8. Vėliau kiekvienas blokelis apdorojamas ir koduojamas atskirai.

(2) Visų blokelio 8×8 elementų (taškų) reikšmės centruojamos, atimant 128.

(3) Blokeliui 8×8 taikoma DKT, būtent:

$$S_{v,u} = \frac{1}{4} C_u C_v \sum_{x=0}^{7} \sum_{y=0}^{7} s_{x,y} \cos\frac{(2x+1)u\pi}{16} \cos\frac{(2y+1)v\pi}{16}, \quad \textbf{(2.31)}$$

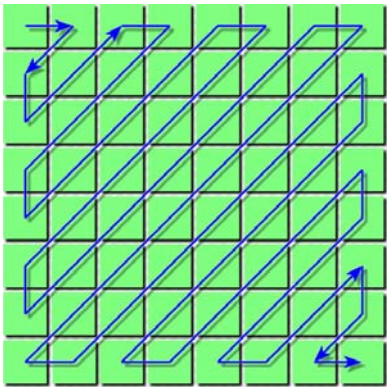
$C_0 = \frac{1}{\sqrt{2}}$; $C_u = C_v = 1$, kai $u \neq 0$, $v \neq 0$.

(4) Gauti spektriniai DKT koeficientai kvantuojami, panaudojant specialiai parinktas kvantavimo matricas Q, t.y. kiekvienas DKT matricos koeficientas $S_{v,u}$ yra dalinamas iš pasirinkto kvantavimo matricos elemento $Q_{v,u}$ reikšmės, būtent:

$$S'_{v,u} = round\left(\frac{S_{v,u}}{Q_{v,u}}\right), \quad \textbf{(2.32)}$$

čia: $Q_{v,u}$ – yra (v, u)–asis kvantavimo matricos elementas. Prarandamos informacijos (būtent šiame etape) kiekis priklauso nuo panaudotos matricos Q vidinės struktūros.

(5) Kvantuoti spektriniai DKT koeficientai (kiekvienam blokeliui 8×8) nuskaitomi zigzagine tvarka (2.5 pav.); be to, saugomos ne absoliučios koeficiento $S_{0,0}$ reikšmės, o tų reikšmių skirtumai, nuosekliai apeinant blokelius 8×8.



2.5 pav. Zigzaginis kvantuotų spektrinių DKT koeficientų nuskaitymas

(6) Nuskaityti spektriniai koeficientai papildomai apdorojami, taikant aritmetinį (arba Huffmano) kodavimą.

(7) Koduoti duomenys išsaugomi.

Vaizdo dekodavimas atliekamas, vykdant tuos pačius veiksmus atvirkščia tvaka (su nežymiais pakeitimais), būtent: kvantavimą keičiant dekvantavimu $R_{v,u} = S'_{u,v} \cdot Q_{u,v}$ ir vietoj DKT imant ADKT:

$$r_{y,x} = \frac{1}{4} C_u C_v \sum_{u=0}^{7} \sum_{v=0}^{7} S_{v,u} \cos\frac{(2x+1)u\pi}{16} \cos\frac{(2y+1)v\pi}{16}; \qquad \textbf{(2.33)}$$

čia $r_{x,y}$ – atkurto vaizdo elementas (taškas).

JPEG kodavimo standartas leidžia pasiekti aukštą vaizdo suspaudimo efektą (10 – 20 kartų), tuo pačiu išsaugant gerą (vizualinę) atkurto vaizdo kokybę. Dėl šių savybių JPEG šiuo metu yra bene populiariausias skaitmeninių vaizdų kodavimo standartas.

2.2.3 Hiperboliniai vaizdų filtrai

Hiperbolinio vaizdų filtravimo metodas yra gana paprasta bei efektyvi vaizdų suglaudinimo priemonė. Metodas gali būti taikomas tiek daugiamačiams, tiek vienmačiams nespalvotiems vaizdams. Metodas grindžiamas šiomis prielaidomis:

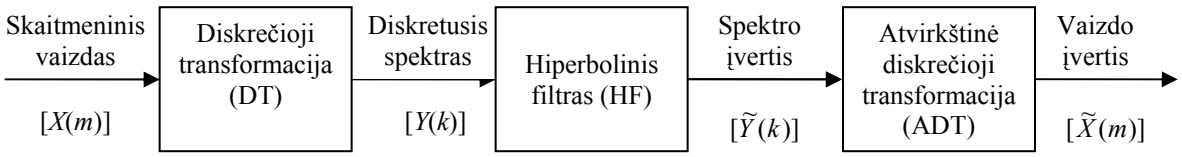
- ✓ Žmogaus akis nėra jautri aukštoms vaizdą sudarančioms harmonikoms;
- ✓ Vaizdą sudarančių harmonikų amplitudės turi tendenciją mažėti, didėjant jų eilės numeriams.

Bendra hiperbolinio vaizdo filtro veikimo schema pateikta 2.6 paveikslėlyje. Pastebėsime, jog hiperboliniam vaizdo filtravimui gali būti naudojama bet kuri diskrečioji transformacija (DT), duodanti kompaktišką nenulinių spektrinių koeficientų išdėstymą DKT, VAT_h, HT ir kt. Svarbu, kad DT baziniai vektoriai būtų išdėstyti pagal dažnį.

Imkime dvimatį skaitmeninį vaizdą $[X(m)]$, $m = (m_1, m_2)$, $m_i = 0, 1, ..., N-1$, su visais $i = 1, 2$. Jo dvimatį diskretųjį spektrą pažymėkime $[Y(k)]$, $k = (k_1, k_2)$, $k_i = 0, 1, ..., N-1$, su visais $i = 1, 2$. Hiperbolinio vaizdo filtro veikimas nusakomas išraiška:

$$\widetilde{Y}(k) = \widetilde{Y}(k_1, k_2) = \begin{cases} Y(k_1, k_2), & \bar{k}_1 \cdot \bar{k}_2 \le M, \\ 0, & \bar{k}_1 \cdot \bar{k}_2 > M; \end{cases} \qquad \textbf{(2.34)}$$

čia: $\bar{k}_i = \max\{k_i, 1\}$, $i \in \{1, 2\}$; M – hiperbolinio filtro lygis dvimatėje erdvėje, $1 \le M \le N^2$.



2.6 pav. Hiperbolinio vaizdų filtro veikimo schema

Kaip matome, spektriniai koeficientai, kurių indeksų sandauga didesnė už filtro lygį, yra atmetami (suspaudimo efektas). Atkuriant vaizdą, atmestieji spektriniai koeficientai keičiami nuliais – gaunamas vaizdo įvertis (atkurtas vaizdas) $[\widetilde{X}(m)]$.

Pastebėsime, jog panaši hiperbolinio vaizdų filtravimo schema gali būti taikoma taipogi vienmačiams bei trimačiams vaizdams, aišku, atitinkamai parenkant vienmatę arba trimatę diskrečiąją transformaciją. Hiperbolinio vaizdo filtras leidžia suspausti vaizdą 5 – 10 kartų, kartu išsaugant gana neblogą atkurto vaizdo kokybę. Pastebėsime, jog vaizdo suspaudimo efektyvumas stipriai susijęs su vaizdo glodumu. Kuo vaizdas glodesnis, tuo geresnis bendras hiperbolinio filtravimo rezultatas.

Pastaruoju metu vis didesnio IT specialistų dėmesio susilaukia progresyvaus vaizdų kodavimo (glaudinimo) požiūris, kai atkuriamo vaizdo kokybė gerinama nuosekliai pasiunčiant papildomus informacijos bitus.

Šį požiūrį geriausiai atstovauja neseniai sukurtas EZW (Embedded Zero-Tree Wavelet) algoritmas, [7].

2.2.4 Fraktalinės vaizdų suglaudinimo procedūros

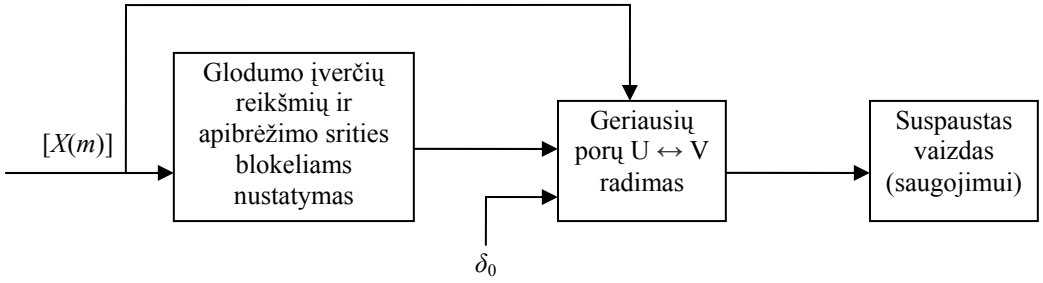
Fraktalas yra sudėtinis geometrinis darinys, kurio atskiri fragmentai yra panašūs arba identiški visumai arba kitiems fragmentams. Pagrindinė fraktalų bendra savybė yra panašumas į save, t.y. išdidinta maža geometrinės struktūros dalis atrodo identiška didesnei daliai.

Fraktalinis vaizdų kodavimas yra gana nauja ir perspektyvi skaitmeninių vaizdų apdorojimo technologija. Deja, didelės skaičiavimų, siejamų su vaizdo kodavimo etapu, apimtys (asimetrijos problema) ir šiandien tebėra rimta kliūtis šios technologijos taikymui [9].

Bazinė fraktalinio dvimačių nespalvotų vaizdų kodavimo (suspaudimo) idėja yra tokia: apdorojamas vaizdas $[X(m)]$ skaidomas dviem lygiais – į reikšmių srities blokelius $[U(m)]$, kurių dydis 8x8 arba 4x4 ir į apibrėžimo srities blokelius $[V(m)]$, kurių dydis atitinkamai 16x16 arba 8x8. Reikšmių srities blokeliai yra nepersidengiantys ir apima visus vaizdo $[X(m)]$ elementus, tuo

tarpu kai apibrėžimo srities blokeliai gali persidengti ir nebūtinai apima kiekvieną vaizdo elementą. Fraktalinio vaizdo kodavimo esmė – kiekvienam reikšmių srities blokeliui [$U(m)$] reikia rasti apibrėžimo srities blokelį [$V(m)$] tokį, kad atstumas (vidutinė kvadratinė paklaida) $\delta = \delta(U,V)$ būtų minimalus. Bendram rezultatui pagerinti, apibrėžimo srities blokeliams papildomai taikomos transformacijos (posūkis, atspindys, kontrastiškumo keitimas). Akivaizdu, jog geriausių porų [$U(m)$] ↔ [$V(m)$] paieškai skirtos laikinės sąnaudos yra didžiulės (tenka tikrinti ~0,5 · 10^9 porų, vaizdams 256x256; ~8,1 · 10^9 porų, vaizdams 512x512). Labai daug pastangų dėta ir dedama, kad sumažinti geriausių porų [$U(m)$] ↔ [$V(m)$] paieškos laiką. Naudojami įvairūs požiūriai – keičiamas blokelių dydis, apsiribojama lokalizuota paieška ir pan.

Vaizdo dekodavimo etapas yra kur kas spartesnis ir paprastesnis – kiekviena atrinkta pora [$U(m)$] ↔ [$V(m)$] nusako suspaudžiančiąją afiniąją transformaciją, o pastarųjų visuma apibrėžia iteruotųjų funkcijų sistemą (IFS).



2.7 pav. Fraktalinio vaizdų kodavimo schema – būtinos vaizdų panašumo sąlygos taikymas

2.3 Vaizdų suglaudinimo metodai be informacijos praradimo

Vaizdų suglaudinimo metodai be informacijos praradimo naudojami tuomet, kai informacijos praradimas netoleruotinas arba patys vaizdai yra specifiniai (grafikai, brėžiniai ir kt.). Natūralios kilmės vaizdams šie metodai leidžia pasiekti 2 – 3 kartų suspaudimo laipsnį, o specifiniams vaizdams gaunami kur kas geresni rodikliai. Trumpai aptarsime kelis populiarius tokio tipo suspaudimo metodus: RLE ir LZW.

2.3.1 RLE kodavimo metodas

Vienas iš jų yra RLE – labai paprastas duomenų suspaudimo metodas, paremtas idėja, jog prieš kiekvieną koduojamą reikšmę galima nurodyti jos pasikartojimų iš eilės skaičių. Iš čia kilęs ir pats metodo pavadinimas (RLE – Run Length Encoding) [3]. Tarkime, norime užkoduoti simbolių seką „ABCCCBBAAABBB“.

Turime tris kartus iš eilės pasikartojantį simbolį „C“, su kartus – „B“, tris kartus – „A“ ir tris kartus – „B“. Užkoduota seka atrodytų taip „ABC3BAA3B3“. Suspaudimo laipsnis šiuo atvejus bus: $\beta = 13 / 10 = 1,3$.

RLE kodavimo metodas tinka koduoti tiems vaizdams, kuriuose yra daug vienodų iš eilės einančių elementų (reikšmių), tačiau kai vaizdas įvairesnis, galima gauti ir žymiai blogesnius rezultatus nei išvis nenaudojant suglaudinimo. Siekiant to išvengti, įvedamas specialus „žymėjimas“ nepasikartojančių reikšmių sekoms. Tokiai žymei galima naudoti šešioliktainį skaičių 0x00. Šioje vietoje realizacija gali būti įvairi atsižvelgiant į glaudinimo duomenų specifikaciją.

Praktikoje naudojama keletas įvairių šio metodo modifikacijų: Microsoft RLE, CompuServe RLE ir kt. Šis metodas naudojamas šiuose vaizdų ir ne tik kodavimo formatuose: BMP, PDF, TIFF.

2.3.2 LZW kodavimo metodas

Šis metodas sudėtingesnis už RLE, bet ne mažiau populiarus. Pirmąsyk jis publikuotas 1977 m. (Lempel Ziv), o vėliau 1984 m. papildytas (Teryy Welch). Metodas remiasi pasikartojančių simbolių sekų žodyno sudarymu, o suspaudimo efektas gaunamas, vietoj pasikartojančių sekų išsaugant tik sekos numerį, kuriuo ji užregistruota žodyne. Ypatingai naudinga algoritmo savybė ta, jog žodyną visai nebūtina išsaugoti – jis gali būti atstatytas išskleidimo metu [4].

Abu pastarieji algoritmai gana paprasti ir pakankamai greiti, egzistuoja ne viena komercinė jų realizacija, tačiau jų panaudojimą apsunkina patentai. Šis metodas (vidutiniškai) pasiekia didesnį nei „2 kartai“ suspaudimo laipsnį. LZW panaudojamas šiuose grafinių vaizdų kodavimo formatuose: GIF, TIFF, PCX.

2.4 Dekoduoto vaizdo kokybės įvertinimas

Akivaizdu, kad būtina įvertinti dekoduotą vaizdą ir nustatyti, kiek jis skiriasi nuo pradinio vaizdo. Tačiau vaizdo kokybės įvertinimas yra subjektyvus ir nėra paprasta įvertinti tą skirtumą.

Sistemoje naudojami du skirtumo įvertinimo būdai: vidutinė kvadratinė paklaida pagal vaizdo elementus (δ) ir didžiausias santykis signalas/triukšmas (PSNR, peak signal-to-noise ratio).

Vidutinė kvadratinė paklaida pagal vaizdo elementus apskaičiuojama taip:

$$\delta = \delta(X_1, X_2) = \sqrt{\frac{1}{N^2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}(X_1(i,j) - X_2(i,j))^2}\ ; \qquad \textbf{(2.35)}$$

čia: X_1, $X_2 \in S^2$, $N = 2^n$, $n \in \mathbf{N}$. Kuo mažesnė vidutinės kvadratinės paklaidos pagal vaizdo elementus reikšmė, tuo geresnė yra dekoduoto vaizdo kokybė.

PSNR – tai standartinis kiekybinis suspausto vaizdo iškraipymų įvertinimas. PSNR matuojama decibelais pagal logaritminę skalę. PSNR apskaičiuojama taip:

$$PSNR = 20 \cdot \lg\frac{2^p - 1}{\delta}, \qquad \textbf{(2.36)}$$

čia p – bitų, skirtų vaizdo elemento reikšmei koduoti, skaičius.

Jei dekoduoto vaizdo PSNR reikšmė yra didesnė negu 35 dB, žmogaus akis neatskiria šio vaizdo nuo originalo. Jei PSNR yra tarp 30dB ir 35dB, dekoduoto vaizdo kokybė puiki; jei PSNR yra tarp 25dB ir 30dB, vaizdo kokybė labai gera; jei PSNR yra mažesnė negu 20dB, vaizdo kokybė yra nepatenkinama, [12].

2.5 Progresyviojo vaizdų kodavimo idėja – EZW algoritmas

EZW algoritmas įgyvendina progresyviojo vaizdų kodavimo idėją – vaizdas koduojamas bitų srautu, kuriame bitai išrikiuoti vaizdo tikslumą didinančia tvarka. Iš pradžių EZW algoritmas buvo skirtas darbui su dvimačiais skaitmeniniais signalais, tačiau, kaip parodė vėlesni tyrimai, jis gali būti sėkmingai naudojamas ir kitokio matavimo vaizdams (signalams) koduoti.

Trumpai pristatysime šio algoritmo veikimo principą dvimačiu atveju. Algoritmas „dirba“ spektrinėje srityje, t.y. pirmiausia dvimatis vaizdas diskrečiosios bangelių transformacijos (tarkim HT) pagalba pervedamas į spektrinę sritį. Vaizdo (signalo) glaudinimo metu koduojami diskrečiajame vaizdo spektre esantys nuliniai ir izoliuoti medžiai bei reikšmingi spektriniai koeficientai. Nulinis medis – tai su tam tikru spektriniu koeficientu (šaknimi) susijęs medis, kurio visos viršūnės, įskaitant ir medžio šaknį, yra mažesnės arba lygios (absoliutine reikšme) pasirinktam slenksčiui. Reikšmingas spektrinis koeficientas – toks koeficientas, kurio reikšmės modulis yra didesnis už pasirinktą slenksčio reikšmę.

Pagrindiniai EZW algoritmo etapai:

(1) Nustatomas pradinis koduojamo vaizdo [$X(m_1, m_2)$] spektro [$Y(k_1, k_2)$] skenavimo slenkstis: $t_0 = 2^{r_{\max}}$, kur $r_{\max}$ yra sveikoji skaičiaus $\log_2(\max\{|Y(k_1, k_2)|\})$ dalis. Realizuojant algoritmą, slenkstis nuosekliai mažinamas, t.y. $t_k = t_0 / 2^k$, $k = 1, 2, \ldots, r_{\max}$.

(2) Su kiekviena slenksčio reikšme t_k atliekamas vaizdo HT spektro skenavimas panaudojant specialias (Morton) kreives (2.8 pav., b). Skenavimo metu spektro koeficientai $Y(k_1, k_2)$ yra užkoduojami (2.8 pav., a): simboliu „*P*“ (jei koeficientas didesnis nei nurodytas slenkstis t_k), simboliu „*N*“ (jei koeficientas mažesnis už neigiamą slenkstį $-t_k$), simboliu „*T*“ (jei koeficientas yra nulinio medžio šaknis) arba simboliu „*Z*“ (jei koeficientas absoliutine reikšme yra mažesnis nei slenkstis t_k, bet medis nėra nulinis). Nurodant, ar spektrinis koeficientas yra nulinio medžio viršūnė, ar izoliuotas medis, išvengiama viso medžio skenavimo. Ši informacija saugoma pagrindiniame sąraše (spektro skenavimo tvarkos sąrašas). Be to, koeficientai, užkoduoti simboliais „*P*“ ir „*N*“, yra šalinami iš spektro (spektre įrašomi nuliai) ir kaupiami FIFO masyve. Šiame masyve sukauptos reikšmės koduojamos dvejetainiais kodais, kurie saugomi papildomame sąraše (spektro koeficientų reikšmių sąrašas).

(3) Antrasis punktas iteruojamas iki $t_k \geq t_{\min}$.

Vaizdo spektro dekodavimas atliekamas, remiantis pagrindiniais ir papildomais sąrašais tokiu būdu:

(1) Naudojant papildomą sąrašą, kuriame saugomos spektro koeficientų reikšmės dvejetaine forma, atkuriamos spektro reikšmės dešimtaine forma.

(2) Atkurtos dešimtainės spektro koeficientų reikšmės iš eilės statomos į atkuriamą spektrą naudojant pagrindinį sąrašą, kuriame saugoma spektro skenavimo tvarka.

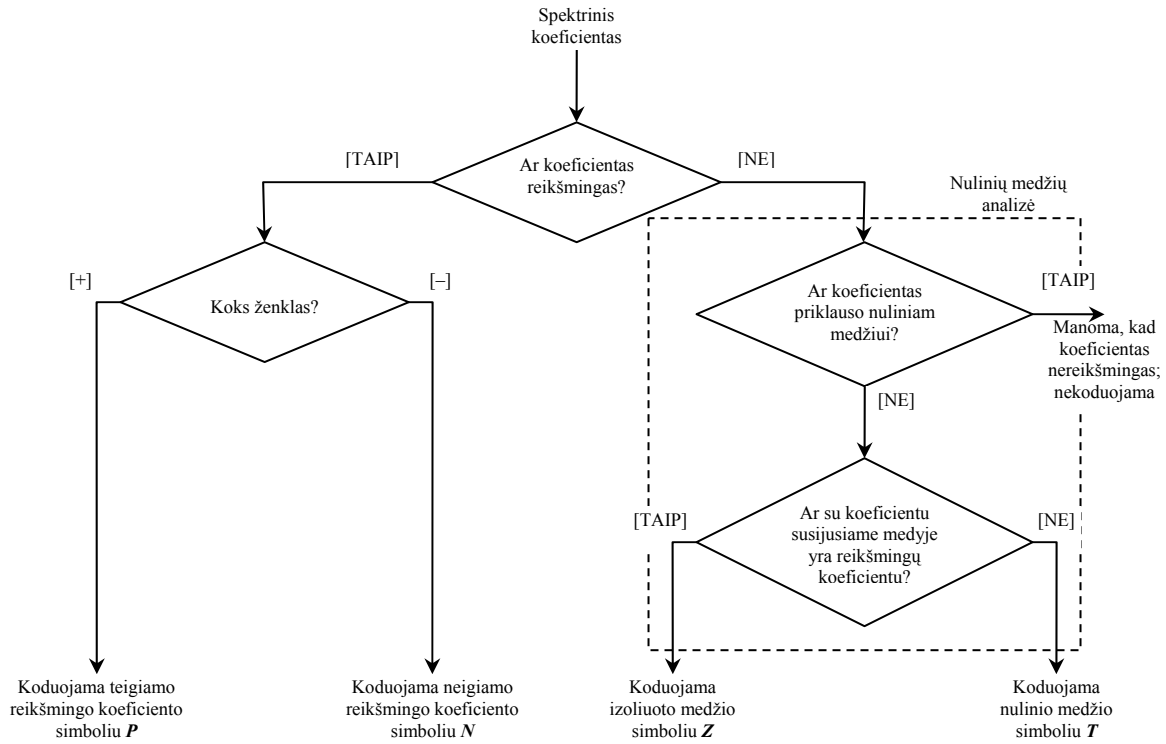
(3) Pirmasis ir antrasis punktai iteruojami kol baigiasi pagrindinio sąrašo elementai (visų iteracijų skenavimo tvarka) arba pagal vartotojo pateiktą pabaigos kriterijų.

Svarbiausi EZW algoritmo privalumai yra šie:

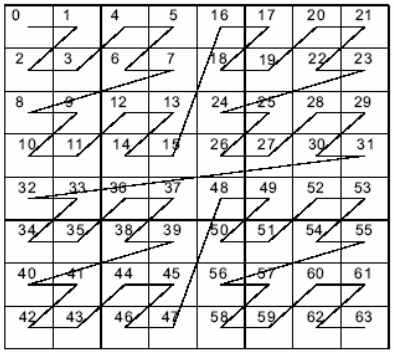
- ✓ Tiksli atkuriamo vaizdo kiekybinė kontrolė – vartotojas visada gali pasirinkti fiksuotą bitų, reikalingų vaizdo elementams (taškams) koduoti, skaičių;
- ✓ Algoritmas gana paprastas ir universalus – tinka įvairioms vaizdų klasėms apdoroti.

Atsiribodami nuo detalaus EZW algoritmo darbo aprašymo, pastebėsime, jog kodavimo rezultate gaunamas bitų srautas papildomai apdorojamas, taikant aritmetinį (be informacijos praradimo) kodavimą.

Pagrindinis EZW algoritmo trūkumas – daugkartinis potencialių nulinių medžių skenavimas (dėl slenksčio reikšmės nuoseklaus mažinimo), t.y. kiekvieną kartą sumažinus slenksčio reikšmę, iš naujo atliekamas diskrečiajame apdorojamo vaizdo spektre esančių nulinių medžių skenavimas ir analizė. Tai daug laiko reikalaujančios procedūros.



(a)



(b)

2.8 pav. Spektro skenavimo schema (EZW): (a) simbolio priskyrimo spektriniam koeficientui schema; (b) spektro skenavimo tvarka (Morton seka)

Darbe EZW algoritmas realizuojamas pasitelkiant diskrečiąją Haaro bangelių transformaciją. Specifinės Haaro bangelių savybės leido suformuoti ir išplėtoti naują progresyvaus vaizdų kodavimo versiją – lokaliai progresyvųjį skaitmeninių vaizdų glaudinimą, kai papildomų informacijos bitų pagalba gerinama ne viso atkuriamo vaizdo kokybė, o tiksliniai parinktų atkuriamų vaizdo fragmentų kokybė.

2.6 Diskrečiosios Haaro (bangelių) transformacijos savybių bei skaičiavimo algoritmų analizė

2.6.1 Specifinės HT koeficientų savybės – lokalizavimas erdvėje

Tarkime, kad [$X(m)$] (m = 0, 1, …, $2^n - 1$) yra vienmatis skaitmeninis vaizdas, o [$Y(k)$] – šio vaizdo diskretusis Haaro spektras. Tada:

(1) Bet kuris spektrinis koeficientas $Y(k) = Y(2^{n-s} + l)$ ($s \in \{1,2...,n\}$, $l \in \{0,1,...,2^{n-s} - 1\}$) yra susijęs ne su visu vaizdu [$X(m)$], o tik su jo fragmentu $X_k = (X(l \cdot 2^s), X(l \cdot 2^s + 1),..., X((l+1) \cdot 2^s - 1))^T$, t.y. šis ir tik šis fragmentas apibūdina skaitinę koeficiento $Y(k)$ reikšmę. Beje, fragmento X_k dydis yra lygus 2^s.

(2) Su spektriniu koeficientu $Y(k)$ ($k = 2^{n-s} + l$) galima susieti medį, kurio viršūnės (spektriniai koeficientai $Y(k^*)$, $k^* \in \bigcup_{t=1}^{s-1} \{2^t k, 2^t k + 1,..., 2^t (k+1) - 1,\}$) yra susiję su tuo pačiu fragmentu X_k, tiksliau su šio fragmento poaibiais (2.9 pav.); koeficientas $Y(k)$ dažnai vadinamas medžio šaknimi.

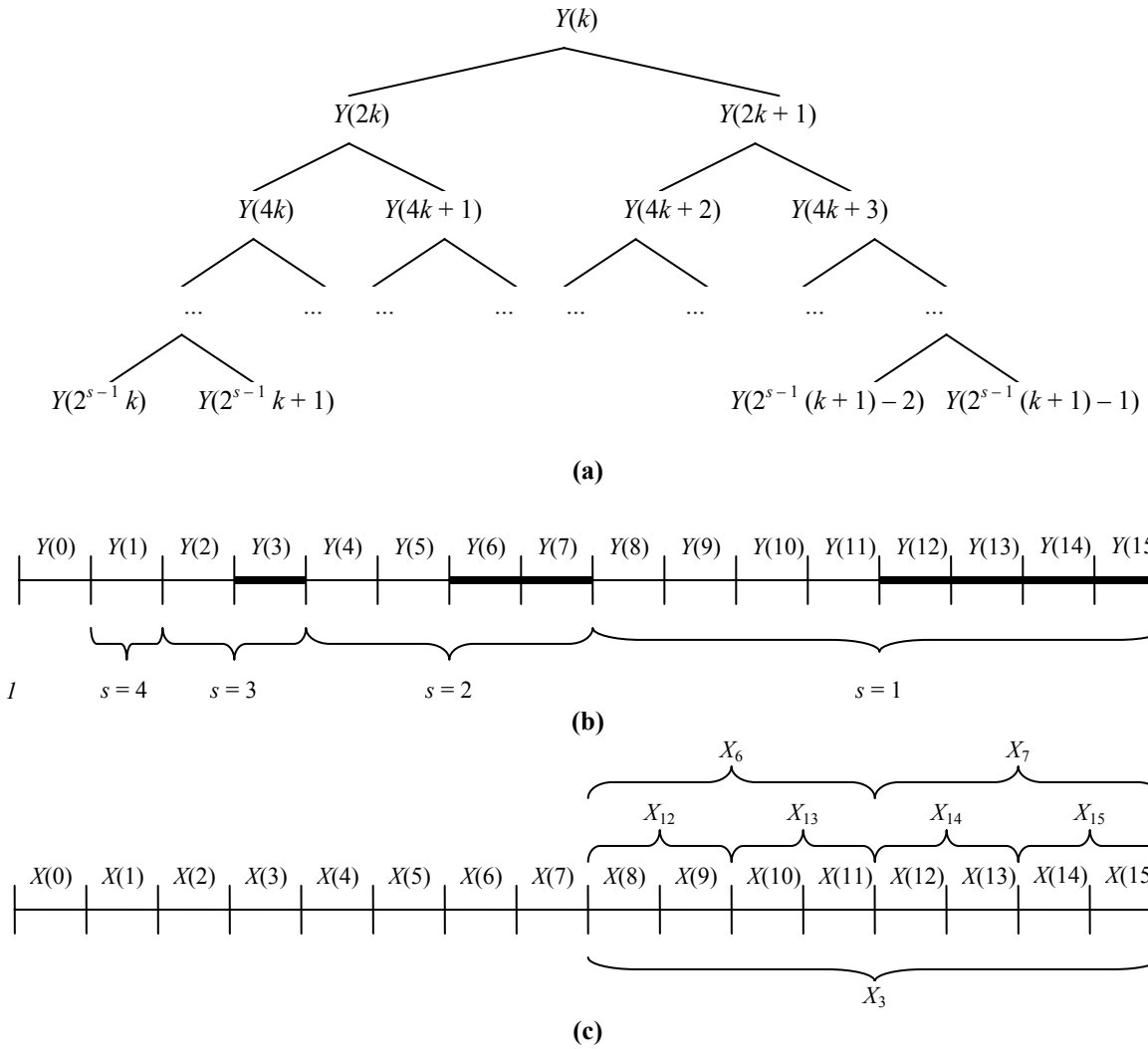
Būtent šios dvi spektrinių Haaro koeficientų savybės sudarė teorines prielaidas naujos lokaliai progresyviojo skaitmeninių vaizdų kodavimo idėjos įprasminimui ir įgyvendinimui.

Dvimatės HT elementai (spektriniai koeficientai) pasižymi analogiškomis savybėmis, kaip ir vienmatės HT atveju (2.10 pav.).

2.6.2 Specifinės HT koeficientų savybės – nulinių medžių egzistavimas

Matome, jog su atskirais diskrečiojo spektro koeficientais susiję medžiai (ypač dvimačiu atveju) turi pakankamai daug viršūnių (spektrinių koeficientų). Vienmačiu atveju medis su šaknimi $Y(k) = Y(2^{n-s} + l)$, $s \in \{1,2,...,n\}$, $l \in \{0,1,...,2^{n-s} - 1\}$ turi $2^s - 1$ viršūnių. Dvimačiu atveju medį su šaknimi $Y(k_1, k_2)$ ($k_i = 2^{n-s_i} + l$, $i = 1,2$) sudaro $(4^{\min\{s_1, s_2\}} - 1)/3$ viršūnės.

EZW atveju, nulinių medžių diskrečiajame apdorojamo vaizdo spektre ieškoma nuosekliai mažinant slenksčio reikšmę ir kiekvienam iš šių atvejų atliekant medį sudarančių viršūnių (spektrinių koeficientų) analizę – palyginimą.

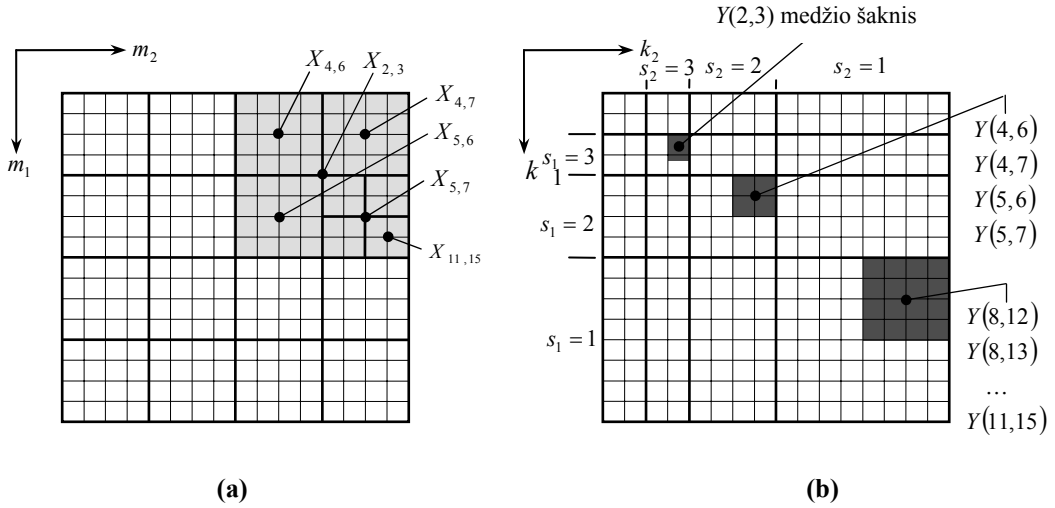


2.9 pav. Geometrinė spektrinių HT koeficientų interpretacija: (a) su koeficientu *Y(k)* susijusio medžio struktūra; (b) koeficientas *Y*(3) ir jį atitinkantis medis (paryškintas; *N* = 16); (c) vaizdo fragmentai X_{k^*}, su kuriais susiję medžio šaknis *Y*(3) bei medžio viršūnės $Y(k^*)$ (*N* = 16)

2.7 Naujas diskrečiojo Haaro spektro apskaičiavimo vaizdo fragmentams algoritmas

2.7.1 Vienmatis atvejis

Tarkime, kad $X = (X(0), X(1),..., X(N-1))^T$ ir $Y = (Y(0), Y(1),..., Y(N-1))^T$ yra, atitinkamai, vaizdo vektorius ir jo diskrečioji Haaro transformacija. Nagrinėsime spektro



2.10 pav. Haaro spektro savybių grafinė interpretacija: (a) vaizdas [$X(m_1, m_2)$], $N = 16$; spektrinis koeficientas $Y(2, 3)$ susijęs su vaizdo fragmentu $X_{2,3}$; (b) koeficientui $Y(2, 3)$ priskirtas medis ($\Im_2 = \Im_2(1) \cup \Im_2(2) = \{4, 5, 8, 9, 10, 11\}$, $\Im_3 = \Im_3(1) \cup \Im_3(2) = \{6, 7, 12, 13, 14, 15\}$; **čia** $\Im_{k_i}(t) = \{2^t \cdot k_i, 2^t \cdot k_i + 1, ..., 2^t \cdot (k_i + 1) - 1\}$, **su visais** $t = 1,2,..., s_i - 1$, $i = 1,2$.)

koeficientą $Y(k)$ ($(k = 2^{n-s} + l) \in \{2,3,..., N-1\}$), susijusį su vaizdo fragmentu $X_k = [X_k(m)]$, $m = 2^s \cdot l, 2^s \cdot l + 1, ..., 2^s \cdot l + 2^s - 1$.

Tegul $Y^{(k)} = (Y^{(k)}(0), Y^{(k)}(1), \ldots, Y^{(k)}(2^s - 1))^T$ žymi fragmento X_k diskretųjį Haaro spektrą. Tada, pasinaudodami Haaro bangelių savybėmis galime efektyviai apskaičiuoti fragmento spektro vektoriaus $Y^{(k)}$ koeficientus, būtent:

$$Y^{(k)}(0) = \overline{X_k}, \tag{2.37}$$

$$Y^{(k)}(i) = \sqrt{2^{n-s}} \cdot Y(k^*), \; i = 1, 2, \ldots, 2^s - 1, \tag{2.38}$$

$$k^* \in \{k\} \cup \Im_k, \; \Im_k = \bigcup_{t=1}^{s-1} \{2^t k, 2^t k + 1, ..., 2^t k + 2^t - 1\}. \tag{2.39}$$

Atitiktis tarp aibių $\{1, 2, ..., 2^{s-1}\}$ ir $\{k\} \cup \Im_k$ yra fiksuota (abipus vienareikšmė); čia $\overline{X_k}$ nurodo pastovią vaizdo fragmento komponentę, kuri yra randama pasinaudojant rekurentiniu sąryšiu:

$$\overline{X_k} = X_{k^\circ} + (-1)^k \sqrt{2^{n-s-1}} \cdot Y(k^\circ), \tag{2.40}$$

$\overline{X_0} = \overline{X_1} = Y(0)$, $k^\circ = [k/2]$; čia $[x]$ – skaičiaus x sveikoji dalis.

2.7.2 Dvimatis atvejis

Tarkime, kad $[X(m_1,m_2)]$ ir $[Y(k_1,k_2)]$ yra atitinamai dvimatis skaitmeninis vaizdas ir jo diskretus Haaro (bangelių) spektras; $m_1,m_2,k_1,k_2 \in \{0,1,...,N-1\}$, $N=2^n$, $n \in \mathbf{N}$.

Kiekvienas spektrinis koeficientas $Y(k_1,k_2), (k_1,k_2) \notin \{(0,0),(0,1),(1,0),(1,1)\}$, turi labai įdomias (praktiniu požiūriu) savybes, būtent:

(1) Koeficiento $Y(k_1,k_2)$ $(k_i = 2^{n-s_i}+l_i, l_i \in \{0,1,...,2^{n-s_i}\}, i=1,2)$ skaitinę reikšmę nusako išskirtinai vaizdo $[X(m_1,m_2)]$ fragmentas (blokas) $X_{k_1,k_2} = [X(m_1,m_2) \mid (m_1,m_2) \in V_{k_1} \times V_{k_2}]$; čia $V_{k_i} = \{l_i \cdot 2^{s_i}, l_i \cdot 2^{s_i}+1,...,(l_i+1)\cdot 2^{s_i}-1\}$, $i=1,2$.

(2) Spektrinis koeficientų poaibis (medis) gali būti priskirtas koeficientui $Y(k_1,k_2)$, kai $s_1>1$ ir $s_2>1$. Medžio viršūnių (spektrinių koeficientų) aibė $\{Y(k_1^*,k_2^*) \mid (k_1^*,k_2^*) \in \bigcup_{t=1}^{\min\{s_1-1,s_2-1\}} (\Im_{k_1}(t) \times \Im_{k_2}(t))\}$; čia $\Im_{k_i}(t) = \{2^t \cdot k_i, 2^t \cdot k_i + 1,...,2^t \cdot (k_i+1)-1\}$, su visais $t=1,2,...,s_i-1$, $i=1,2$. Pats koeficientas $Y(k_1,k_2)$ dažnai vadinamas medžio šaknimi.

✓ Spektrinis koeficientas $Y^{(k_1,k_2)}(0,0)$, sutampantis su vaizdo bloko X_{k_1,k_2} pastoviąja dedamąja (t.y. $Y^{(k_1,k_2)}(0,0) = \overline{X}_{k_1,k_2}$) apskaičiuojamas panaudojant rekurentinius sąryšius, būtent:

$$\overline{X}_{k_1,k_2} = \overline{X}_{k_1^0,k_2^0} + (-1)^{k_1}\sqrt{2^{n-s_1-1}} \cdot A(k_1^0,k_2^0) + (-1)^{k_2}\sqrt{2^{n-s_2-1}} \cdot B(k_1^0,k_2^0) + (-1)^{k_1+k_2}\sqrt{2^{2n-s_1-s_2-2}} \cdot Y(k_1^0,k_2^0),$$

kai $k_1>1$ ir $k_2>1$; čia $k_i^0 = \lfloor k_i/2 \rfloor$, t.y. $[x]$ – sveikoji skaičiaus x dalis; $i \in \{1,2\}$;

$$A(k_1,k_2) = \begin{cases} Y(k_1,0), & k_2 \le 1 \quad (s_2 = n), \\ Y(k_1,0) + \sum_{r=1}^{n-s_2} (-1)^{l_2^{(r)}} \sqrt{2^{n-s_2-r}} \cdot Y(k_1,\beta_r), & k_2 > 1 \quad (s_2 < n), \end{cases} \tag{2.41}$$

$$B(k_1,k_2) = \begin{cases} Y(0,k_2), & k_1 \le 1 \quad (s_1 = n), \\ Y(0,k_2) + \sum_{r=1}^{n-s_1} (-1)^{l_1^{(r)}} \sqrt{2^{n-s_1-r}} \cdot Y(\alpha_r,k_2), & k_1 > 1 \quad (s_1 < n), \end{cases} \tag{2.42}$$

čia $\alpha_r = \alpha_{r-1}^0$ $(r=2,3,...,n-s_1)$, $\alpha_1 = k_1^0$; $\beta_r = \beta_{r-1}^0$ $(r=2,3,...,n-s_2)$, $\beta_1 = k_2^0$; $< l_1^{(n-s_1)} l_1^{(n-s_1-1)} ... l_1^{(1)} >$ ir $< l_2^{(n-s_2)} l_2^{(n-s_2-1)} ... l_2^{(1)} >$ yra atitinkamai dešimtainiai l_1 ir l_2 išraiškos. Beje,

$\overline{X}_{k_1,0} = \overline{X}_{k_1,1} = \overline{X}_{k_1^0,0} + (-1)^{k_1}\sqrt{2^{n-s_1-1}} \cdot Y(k_1^0,0)$, kai $k_1>1$;

$\overline{X}_{0,k_2} = \overline{X}_{1,k_2} = \overline{X}_{0,k_2^0} + (-1)^{k_2}\sqrt{2^{n-s_2-1}} \cdot Y(0,k_2^0)$, kai $k_2>1$; $\overline{X}_{0,0} = \overline{X}_{0,1} = \overline{X}_{1,0} = \overline{X}_{1,1} = Y(0,0)$.

✓ Spektriniai koeficientai $Y^{(k_1,k_2)}(u,v)$ $(u^2+v^2\neq 0)$ apskaičiuojami tokiu būdu:

$$Y^{(k_1,k_2)}(u,0)=\sqrt{2^{n-s_1}}\cdot A(k_1^*,k_2)\,, \tag{2.43}$$

$$Y^{(k_1,k_2)}(0,v)=\sqrt{2^{n-s_2}}\cdot B(k_1,k_2^*)\,, \tag{2.44}$$

$$Y^{(k_1,k_2)}(u,v)=\sqrt{2^{2n-s_1-s_2}}\cdot Y(k_1^*,k_2^*)\,, \tag{2.45}$$

su visais $u=1,2,...,2^{s_1}-1$ ir $v=1,2,...,2^{s_2}-1$; indeksai k_1^* ir k_2^* sutampa su *u*–uoju ir *v*–uoju elementais sutvarkytame (didėjimo tvarka) rinkinyje $\{k_1\}\cup\Im_{k_1}=\{k_1\}\cup\left(\bigcup_{t=1}^{s_1-1}\Im_{k_1}(t)\right)$ ir $\{k_2\}\cup\Im_{k_2}=\{k_2\}\cup\left(\bigcup_{t=1}^{s_2-1}\Im_{k_2}(t)\right)$.

Verta paminėti, jog operacijų skaičius (sudėtis, daugyba), susiję su tiesioginiu HT spektro vaizdo blokui X_{k_1,k_2} apskaičiavimu (tipinis greitas HT skaičiavimo algoritmas; 2.1.4 skyrelis) yra lygus (kai $s_1=s_2=s$) $\Re_t=2^{s+1}(2^s+\sum_{t=1}^{s}2^{s-t}(s-t+3))$. Tuo tarpu, pasiūlytos procedūros atveju, reikalaujamas operacijų skaičius lygus $\Re_n=(2^s-1)^2+4(n-s)(2^s-1)+10(n-s)-4$. Elementarūs skaičiavimai rodo, jog pastaroji procedūra yra $10\div 15$ kartų spartesnė.

Kadangi lokaliai progresyvaus vaizdų kodavimo idėja yra realizuota bangelių spektrinėje srityje, greitas perėjimas nuo viso vaizdo diskrečiojo Haaro spektro prie pasirinkto vaizdo bloko Haaro spektro, yra labai vertingas (vaidina lemiamą vaidmenį).

3. LOKALIAI PROGRESYVAUS VAIZDŲ KODAVIMO PROGRAMINĖS ĮRANGOS PROBLEMOS IR JŲ SPRENDIMO ANALIZĖ

3.1 Projekto tikslas

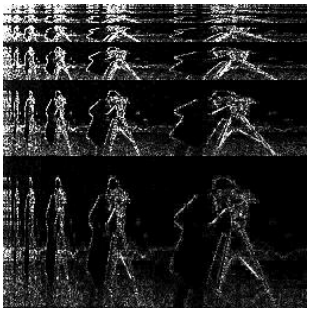
Per pastaruosius dešimt metų skaitmeninių vaizdų (tiek 2D, tiek 3D) kodavimo (arba glaudinimo) srityje pasiektas didžiulis progresas. Tam, kad būtų išlaikytas tolimesnis skaitmeninių vaizdų glaudinimo technologijų progresas, daugelis specialistų pradėjo intensyviai naudoti diskrečiąsias „bangelių“ (angl. *wavelet*) transformacijas. Šiai diskrečiųjų transformacijų šeimai priklauso ir diskrečioji Haaro transformacija. Ši transformacija pasirinkta dėl savo paprastumo bei išskirtinių savybių (viena iš jų – lokalizavimo erdvėje savybė).

Šio projekto tikslas – realizuoti ir ištirti lokaliai progresyvųjį skaitmeninių vaizdų kodavimo/dekodavimo procesą, naudojant diskrečiąją Haaro transformaciją bei naują progresyvųjį EZW kodavimo algoritmą (3.1 pav.). Be to, ištirti naudojamų kodavimo priemonių

„suderinamumą“. Sistemos darbas neturi priklausyti nuo kitų programinių sistemų. Numatyti galimybę, atskirus sistemos komponentus integruoti į kitas sistemas.



(a)



(b)



(c)

Progresyvaus kodavimo pavyzdys

101110101
101001010
001110101
110111011
000111100

+6,4 KB



(c)

101110101
101001010
001110101
110111011

+12,8 KB



(e)

101110101
101001010
001110101

+25,6 KB

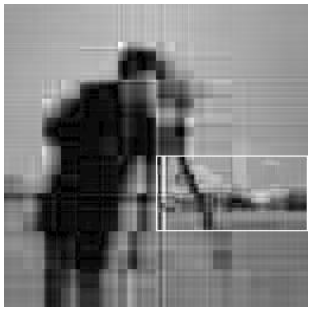


(g)

Lokaliai progresyvaus kodavimo pavyzdys

10101
10010
00101
11011
01100

+0,8 KB



(d)

10101
10010
00101
11011

+1,6 KB



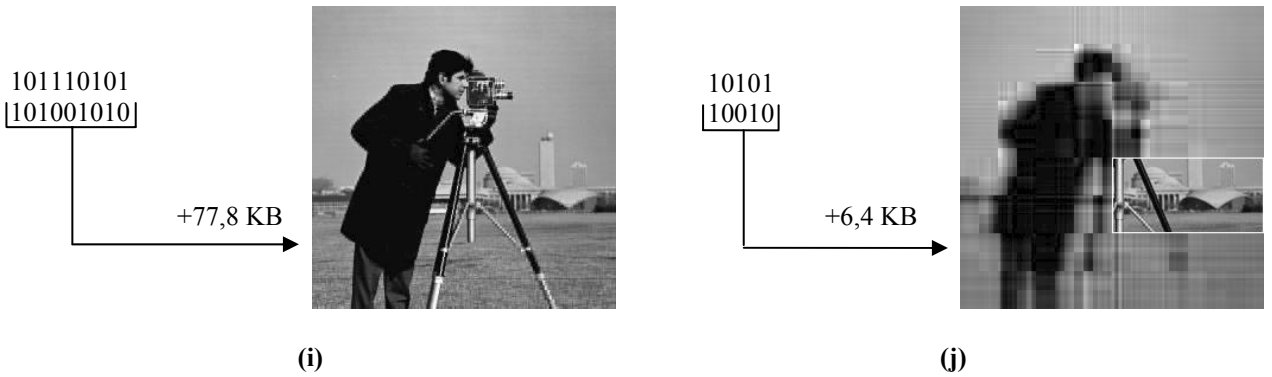
(f)

10101
10010
00101

+3,2 KB



(h)



3.1 pav. Progresyvus (c, e, g, i) ir lokaliai progresyvus (d, f, h, j) vaizdų kodavimas: (a) originalas „*cameraman.bmp*“ 512x512, 256 KB; (b) vaizdo diskretusis Haaro spektras; (c) suspaustas vaizdas β = 40, 6.4 KB; (d) suspaustas vaizdas β = 20, 12.8 KB; (e) suspaustas vaizdas β = 40, fragmentas β = 20, 7.2 KB; (f) suspaustas vaizdas β = 10, 25.6 KB; (f) suspaustas vaizdas β = 40, fragmentas β = 10, 8.8 KB; (g) suspaustas vaizdas β = 5, 51.2 KB; (h) suspaustas vaizdas β = 40, fragmentas β = 5, 12 KB; (i) suspaustas vaizdas β = 2, 128 KB; (j) suspaustas vaizdas β = 40, fragmentas β = 2, 18.4 KB;

3.2 Užduotis

Sukurti programinę įrangą, kurios pagalba realizuojama:

- ✓ Diskrečiosios Haaro transformacijos procedūra (tiesioginė ir atvirkštinė);
- ✓ Progresyvaus EZW kodavimo procedūra;
- ✓ Tarpinių programos duomenų pateikimas:
  - o Koduojamo ir dekoduojamo vaizdo spektrų pateikimas, išsaugojimas, spektro „ryškumo“ keitimas;
  - o Užkoduoto vaizdo apimties pateikimas, duomenų sekos pateikimas ir išsaugojimas (užkoduota byla/vaizdas);
- ✓ Atkurtų vaizdų išsaugojimas (norimu/galimu formatu);
- ✓ Vaizdo kodavimas pagal užduotas vartotojo užklausas (kodavimo slenkstis, kodavimo iteracijų kiekis, spektro koeficientų daugiklis);
- ✓ Dekoduoto vaizdo fragmento detalizavimo procedūra;
- ✓ Vaizdo fragmento išrinkimą palengvinanti sąsaja;
- ✓ Patogi vartotojo sąsaja, suteikianti sistemai dinamiškumo, patogumo bei orientacines laiko sąnaudas.

3.3 Sistemos įgyvendinimo etapai

Projekto realizavimo terminas apima pirmąjį ir antrąjį vykdytojo (asmuo, kuris rengia magistrinį darbą) magistrantūros kursus, t.y. nuo 2006 m. rudens semestro pradžios iki 2008 m. pavasario semestro pabaigos (trukmė ~ 20 mėn.). Kiekvienas sistemos kūrimo etapas pažymėtas

langeliais su kryželiu (☒). Kiekvieno etapo gale gaunamas konkretus produktas, kuris žymimas užtušuotu langeliu (▩). Numatoma darbų seka 1–ojo magistrantūros kurso metu (žr. 3.1 lentelę):

3.1 lentelė. 2006 – 2007 m.m. darbų tvarkaraštis

Nr.	Data (metai, mėnuo) / Darbų seka	2006 m.				2007 m.							
		IX	X	XI	XII	I	II	III	IV	V	VI	VII	VIII
1.	Reikalavimų specifikavimas	☒	☒	☒	☒	▩							
2.	Architektūros specifikavimas					☒	☒	☒	☒	▩			
3.	Projektavimas									☒	☒	☒	▩
4.	Programavimas												☒
5.	Testavimas												
6.	Diegimas												
7.	Tyrimas												

Numatoma darbų seka 2–ojo magistrantūros kurso metu (žr. 3.2 lentelę)

3.2 lentelė. 2007 – 2008 m.m. darbų tvarkaraštis

Nr.	Data (metai, mėnuo) / Darbų seka	2007 m.				2008 m.				
		IX	X	XI	XII	I	II	III	IV	V
1.	Reikalavimų specifikavimas									▨
2.	Architektūros specifikavimas									▨
3.	Projektavimas									▨
4.	Programavimas	☒	☒	▩						▨
5.	Testavimas			☒	▩					▨
6.	Diegimas					▩				▨
7.	Tyrimas					☒	☒	☒	▩	▨

2008 m. paskutinis mėnuo (gegužė) palikti kaip rezervinis mėnuo nenumatytiems atvejams, iškilusiems sistemos įgyvendinimo metu.

3.4 Problemos sprendimas pasaulyje ir Lietuvoje

Per pastarąjį dešimtmetį didelės programinės įrangos kompanijos sukūrė/tebekuria programinės įrangos paketus su įvairiomis grafikos apdorojimo funkcijomis. Be jų egzistuoja ir tokie „maži“ įrankiai, kurių galimybės pilnai tenkina ne tik paprasto vartotojo poreikius, bet ir tam tikros grafikos srities specialistų poreikius. Žemiau pateikiami kelių panaudotų įrankių trumpa apžvalga.

MinImage — tai įrankis, koduojantis skaitmeninius vaizdus bangelių metodu. Pasižymi aiškiu ir greitu skaitmeninių vaizdų kodavimu. Tačiau šis įrankis yra mokamas (20,04 eurų). Tai

vienintelė priežastis, dėl kurios nebuvo galima išmėginti ir pateikti preliminarių glaudinimo įvairiais standartais rezultatų ir pan. Pasak [6] [2], šis įrankis buvo sukurtas vardan skaitmeninių vaizdų glaudinimo bangelių pagrindu. Įrankis geba apdoroti net ir didelių matricų vaizdus, palaiko Bitmap, TIFF, JPEG, PNG, GIF kodavimo standartus. Šiame darbe aktualus HT ir EZW kodavimo būdas nėra realizuotas. Kalbant apie įrankio **MinImage** technines savybes, galima paminėti, jog jis sukurtas C++ programavimo kalbos pagrindu, veikia Microsoft Windows NT platformoje.

Kitas įdomus produktas, tai ReaSoft.com kompanijos **ReaCompressor** įrankis skirtas skaitmeninių vaizdų optimizavimui. Kurdama šį produktą kompanija laikėsi nuostatos, kad neverta pirkti didelių kompanijų daug didesnių programinių paketų (ir brangesnių), tad sukūrė labai paprasto skaitmeninių vaizdų apdorojimo įrankį.

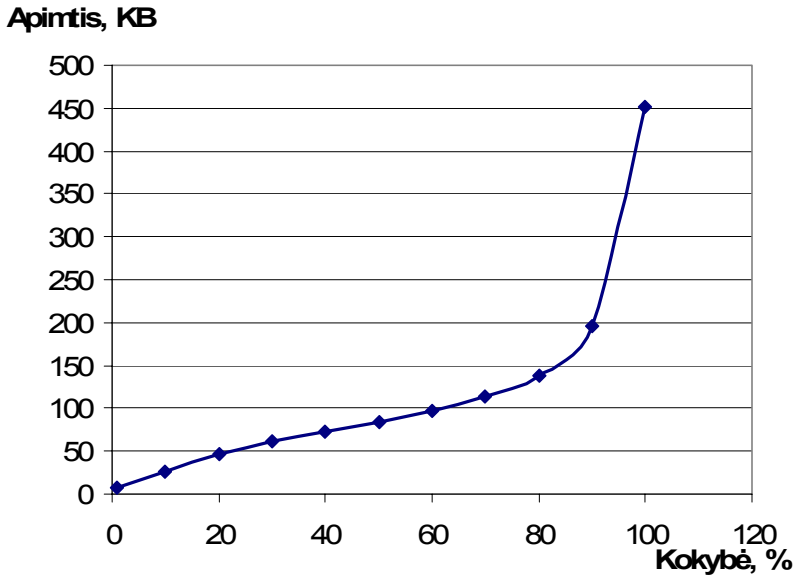
Šis produktas platinamas kaip bandomoji (angl. trial) versija. Licencijos kaina kinta nuo 24,95 iki 99 JAV dolerių. Darbo metu šiuo įrankiu lengvai galima migruoti tarp keturių populiarių kodavimo standartų: JPEG, GIF, PNG ir TIFF. Atpažįsta pakankamai daug kodavimo standartų: TIFF Bitmap (TIF, TIFF), CompuServe Bitmap (GIF), JPEG Bitmap (JPG, JPEG, JPE), PaintBrush (PCX), Windows Bitmap (BMP, DIB, RLE), Windows Icon (ICO), Windows Cursor (CUR), Portable Network Graphics (PNG), Windows Metafile (WMF), Enhanced Windows Metafile (EMF), Targa Bitmap (TGA, TARGA, VDA, ICB, VST, PIX), Portable Pixmap, GreyMap, BitMap (PXM, PPM, PGM, PBM). Produktas skirtas Microsoft Windows šeimos platformoms (95/98/ME/NT/2000/XP).

Žemiau pateikti **ReaCompressor** įrankio bandymo rezultatai: skaitmeninio vaizdo duomenų apimties priklausomybė nuo kokybės ar spalvų gamos (3.3, 3.4 lentelės bei 3.2, 3.3 pav.). Bandymui naudotas BMP formato 1200x900 raiškos vaizdas, kurio apimtis 1,03 MB.

3.3 lentelė. JPEG kodavimo standarto rezultatai

Kokybė, %	Bylos apimtis, KB	PSNR, dB	δ	Kokybė vartotojo akimis
100	450,56	58,774	0,294	Puiki
90	195,44	44,155	1,58	Puiki
80	138,57	41,204	2,2	Puiki
70	113,35	31,762	6,584	Labai gera
60	96,25	38,554	3,012	Labai gera
50	84,73	37,691	3,327	Labai gera
40	73,68	31,156	7,059	Gera
30	61,59	35,821	4,126	Gera
20	46,06	34,37	4,877	Gera (nežymus „raibuliavimas“ Gibsono efektas)
10	26,25	31,702	6,629	Pakankama (nežymi blokinė struktūra)

1	7,65	25,497	13,542	Bloga (blokinė struktūra)



3.2 pav. JPEG formato failo dydžio priklausomybė nuo vaizdo kokybės

Produkto (sistemos), kurio paskirti būtų skaitmeninių vaizdų kodavimas, naudojant Haaro transformaciją bei EZW kodavimo algoritmą, Lietuvoje nėra (nerasta naudojantis visuotinio tinklo paieškos sistema).

3.5 Pagrindinės sistemos projektavimo, programavimo inžinerijos problemos, jų sprendimai

3.5.1 Objektiškai orientuoto programavimo problema ir jos sprendimas

Išanalizavus sistemos reikalavimus, buvo atliktas sistemos projektavimas. Suprojektuotos sistemos aprašas pateiktas 4.3 skyrelyje, o bendras vaizdas 4.2 paveikslėlyje.

Sistemos projektavimas atliktas laikantis objektiškai orientuoto programavimo principų. Kadangi numatomas sistemos tobulinimas ateityje, ji turi būti suprogramuota (sukoduota) objektiškai, komentarų pagalba lengvai skaitoma. Suprogramuota sistema lengvai modifikuojama, dėka sistemos objektiškumo. Matematiniams modeliams pritaikytas aukšto lygio objektiškumas. Pavyzdžiui:

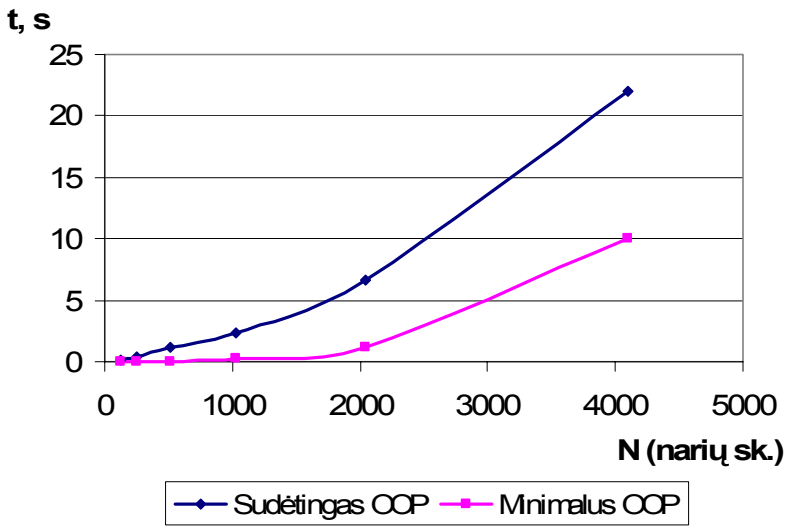
```
Transformacija {
              Spektras1 {
                        Kintamasis1 : double[][]
                        Kintamasis2 : double[][]
```

```
                        gautiNari(i : int, reiksme : double)
                        sumuoti()
                        ...}
}
```

Prisiminkime, jog sistemos paskirtis koduoti ir dekoduoti vaizdus pasirinktais matematiniais metodais. Pagrindiniai reikalavimai sistemai yra greitas vaizdų kodavimas ir optimalus operatyvios atminties naudojimas. Tačiau reikia paminėti, jog pasirinktas objektiško programavimo stilius yra „perteklinis“. Kadangi sistema manipuliuoja dideliais duomenų kiekiais, t.y. didelėmis skaičių matricomis (milijonas elementų ir daugiau), ilgomis simbolių eilutėmis (šimtai tūkstančių simbolių), sistema praranda darbo našumą. Apdorojant 1024x1024 ir didesnės raškos vaizdus, sistema okupuoja beveik visą laisvą operatyvią atmintį, lyginant su kitomis skaitmeninių vaizdų apdorojimo (MinImage, ReaCompressor) programomis kodavimo trukmė nepalyginamai didesnė (šiuo atveju taikoma ir lyginama tik HT metodo realizacijos sparta).

Problema išspręsta minimizuojant sistemos OOP sudėtingumą, t.y. sistemos kodo restruktūrizavimas atliktas tik matematiniams modeliams (HT, EZW). Sistemos sparta (HT metodo) esant sudėtingam ir minimizuotam OOP pateikta 3.3 – ame paveikslėlyje.

Sprendžiant neracionalų operatyvios atminties išnaudojimą, nuspręsta performuoti EZW kodavimo algoritmo Morton skenavimo sekos medžio generavimą. Šiai problemai spręsti buvo pasirinkti dinaminiai sąrašai. Pastarųjų privalumas tas, kad nulinių medžių radimo atveju, iš sistemos atminties pašalinama medžio (didelė ar maža) spektro duomenų šaka. Tokiu būdu atlaisvinama nebenaudojama operatyvi atmintis bei pagreitėja paieška likusiame spektriniame medyje.



3.3 pav. OOP sudėtingumo įtaka sistemos (HT, EZW modeliams) darbo našumui

4. PROGRAMINĖS ĮRANGOS TECHNINĖS/PROJEKTINĖS DOKUMENTACIJOS ESMINIAI ASPEKTAI

4.1 Projekto apribojimai

Sukurta programinė įranga turi veikti kompiuteriuose, kurie tenkina sekančius techninius bei programinės įrangos reikalavimus:

Techniniai asmeninio kompiuterio reikalavimai:

- ✓ Procesorius — Pentium IV 1000 MHz ir spartesnis;
- ✓ Operatyvioji atmintis — 256 MB ir daugiau;
- ✓ Diskinė atmintis — 10 MB;

Programinės įrangos asmeninio kompiuterio reikalavimai:

- ✓ Operacinė sistema — Microsoft Windows XP/2000;
- ✓ Programinė įranga — JAVA 1.5.

Sistema kuriama naudojant JAVA (JAVA 1.5 programų vykdymo aplinka) programavimo kalbą. Sukurta sistema nėra priklausoma nuo jokios kitos programinės įrangos ar sistemų.

Techniniai reikalavimai iškelti orientuojantis į užsakovo naudojamos technikos (kompiuterio) parametrus. Ateityje numatoma galimybė sukurtą programinę įrangą realizuoti ir kitos platformos operacinėse sistemose.

4.2 Pagrindinės sistemos funkcijos

Sukurta sistema geba atlikti skaitmeninio vaizdo kodavimą, dekodavimą, pasirinkto vaizdo fragmento detalizavimą, originalaus ir atkurto bei originalaus ir atkurto su detaliu fragmentu vaizdo kokybės parametrų apskaičiavimu (PSNR, δ) ir kt. Sistema bus nepriklausoma nuo kitų programinių sistemų.

Pagrindinės sistemos funkcijos yra:

- ✓ Skaitmeninio vaizdo kodavimas;
- ✓ Užkoduoto vaizdo dekodavimas (atkūrimas);
- ✓ Tarpinių programos duomenų pateikimas:
  - o Koduojamo ir dekoduojamo vaizdo spektrų pateikimas, išsaugojimas, spektro „ryškumo“ keitimas;

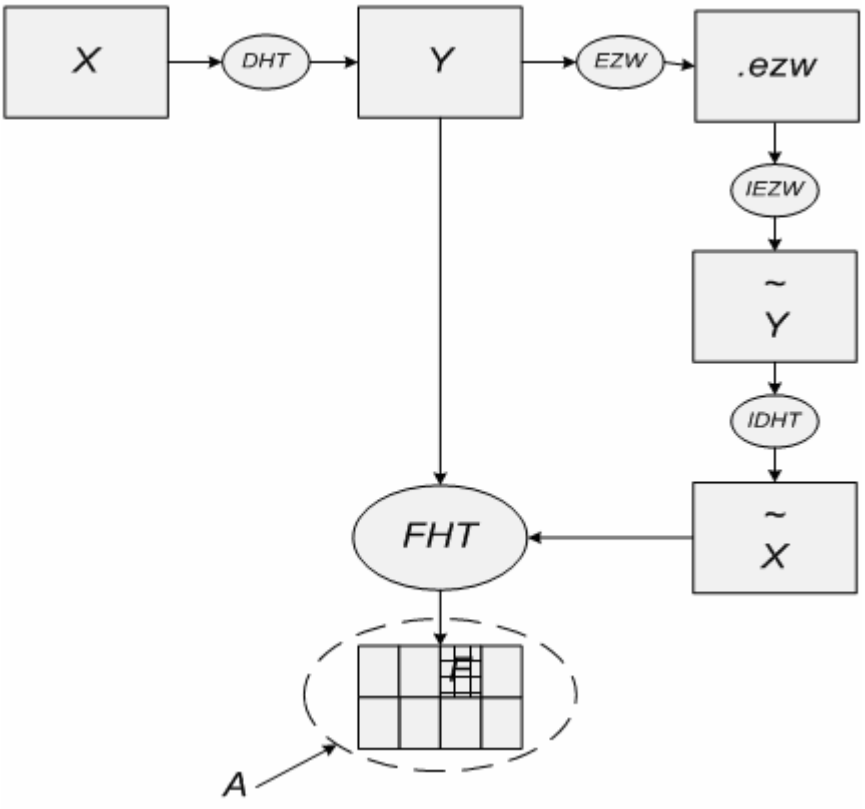
- Užkoduoto vaizdo apimties pateikimas, duomenų sekos pateikimas ir išsaugojimas (užkoduota byla/vaizdas);

✓ Atkurto vaizdo fragmento išrinkimas ir detalizavimas;

✓ Originalaus vaizdo palyginimas su atkurtu vaizdu ir palyginimas su atkurtu vaizdu, kuriame atkurtas detalus fragmentas, parametrų apskaičiavimas;

✓ Atkurtų vaizdų išsaugojimas (norimu/galimu formatu): (BMP, JPEG, GIF, PNG, TIFF, ezw);

✓ Patogi vartotojo sąsaja, suteikianti sistemai dinamiškumo, patogumo, orientacines laiko sąnaudas.



4.1 pav. Skaitmeninio vaizdo apdorojimo (kodavimo, dekodavimo, detalizavimo) veiksmų seka

Sutartiniai žymėjimai:

✓ X — skaitmeninis vaizdas (BMP, JPEG, GIF, PNG, TIFF formato byla);

✓ DHT — diskrečioji Haaro transformacija;

✓ Y — vaizdo spektras;

✓ EZW — EZW kodavimo algoritmas;

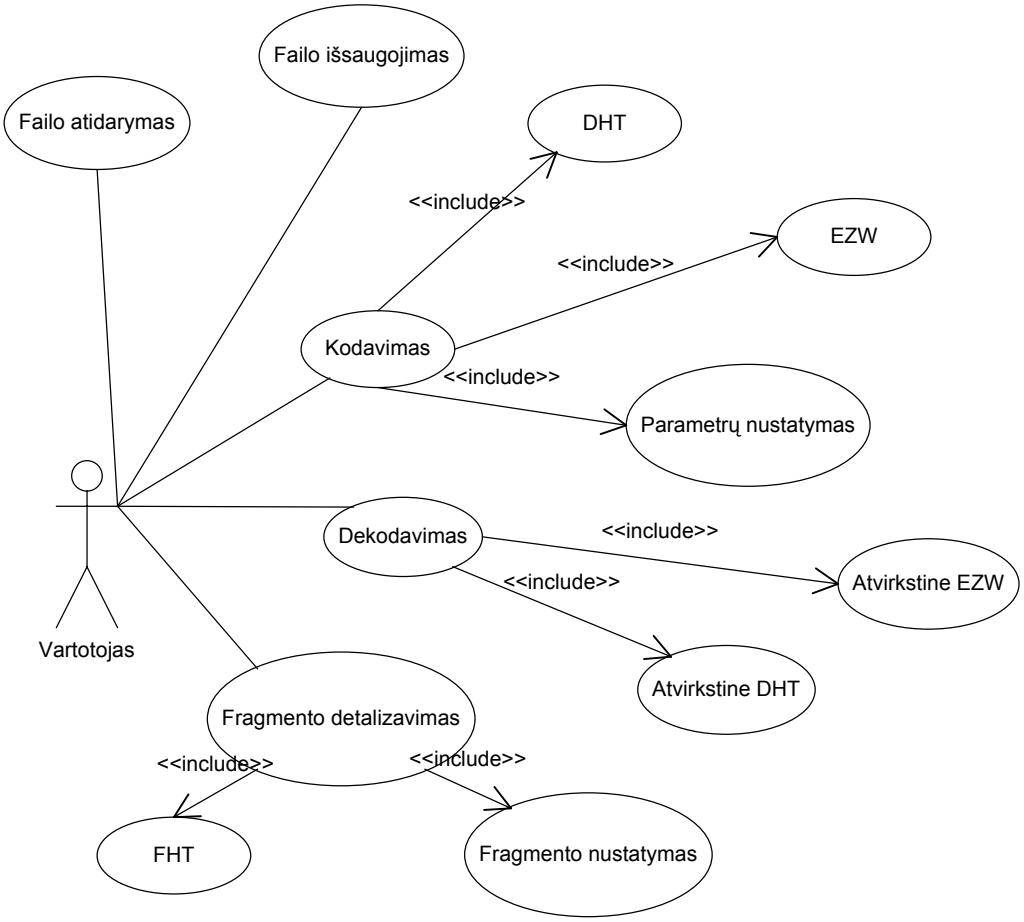
✓ .ezw — suspaustas vaizdas (vaizdas išsaugomas byloje ezw prievardžiu);

✓ IEZW — atvirkštinis EZW glaudinimo algoritmas (dekodavimas);

- ✓ $\widetilde{Y}$ — atkurtas vaizdo spektras;
- ✓ IDHT — atvirkštinė diskrečioji Haaro transformacija;
- ✓ $\widetilde{X}$ — atkurtas skaitmeninis vaizdas ;
- ✓ FHT — greitoji Haar‘o transformacija fragmento apskaičiavimui;
- ✓ A — atkurtas vaizdas su detaliu fragmentu F;
- ✓ ⬭ (vientisa elipsė) — atliekama operacija;
- ✓ ▭ (stačiakampis) — duomenys, gauti rezultatai.

4.3 Sistemos sudėtis

Kaip minėta ankstesniame skyriuje, sistema yra nepriklausoma nuo kitų programinių sistemų. Esant reikalui, sistemos matematiniai moduliai kaip atskiri komponentai gali būti integruoti į kitą sistemą. Sistemos panaudos atvejų modelis matomas 4.2–ame paveikslėlyje.



4.2 pav. Sistemos panaudos atvejų modelis

Panaudos atvejų modelio aktoriai:

✓ Vartotojas — tai vartotojas (sistema valdoma vartotojo sąsaja), kuriam reikalingas koduotas / dekoduotas skaitmeninis vaizdas.

Modelio panaudos atvejai:

✓ Failo atidarymas — atidaromas skaitmenis failas; atidaromi penki pagrindiniai vaizdų formatai: BMP, JPEG, GIF, PNG, TIFF.

✓ Failo išsaugojimas — duomenų išsaugojimas skaitmeninio failo pavidalu; galima išsaugoti pagrindiniais formatais: BMP, JPEG, GIF, PNG, TIFF bei ezw.

✓ Kodavimas — skaitmeninis vaizdas bus užkoduotas naudojant diskrečiąją HT ir EZW algoritmą.

✓ Parametrų nustatymas — nustatomi kodavimo parametrai, t.y. keliomis iteracijomis koduoti duomenis, galutinis kodavimo slenkstis, filtruoti/nefiltruoti spektrą papildomais koeficientais.

✓ DHT — diskrečioji Haaro transformacija taikoma skaitmeniniam vaizdui.

✓ EZW — šis algoritmas yra iteracinis, susidedantis iš dviejų žingsnių (kelio skanavimo ir dešimtainių reikšmių kodavimu dvejetaine forma).

✓ Dekodavimas — funkcija atvirkščia kodavimui. Pirmiausia atliekamas EZW dekodavimas, po to atvirkštinė diskrečioji Haaro transformacija.

✓ Atvirkštinė EZW — dekoduojamas suglaudinto vaizdas, rezultate gaunamas atkurtas vaizdo spektras.

✓ Atvirkštinė DHT — rezultate gaunamas vaizdas artimas pirminiam (originaliam).

✓ Fragmento detalizavimas — atliekamas atkurto (dekoduoto) vaizdo fragmento išskyrimas ir jo atvaizdo detalizavimas;

✓ Fragmento nustatymas — nustatomas atkurto vaizdo fragmentas (atkuriamo vaizdo fragmento indeksų nustatymas).

✓ FHT — tai greitoji Haaro transformacija, kuri apskaičiuoja nurodyto fragmento pirminį vaizdą.

4.4 Sistemai keliami reikalavimai

4.4.1 Funkciniai reikalavimai

- ✓ Skaitmeninės grafinės bylos nurodymas kodavimo ar dekodavimo operacijai:
  - o Išvedamas pranešimas pasirinkus netinkamą bylos formatą;
  - o Bylos nurodymo lange skaitmeninės bylos filtruojamos pagal bylos formatą;

- ✓ Skaitmeninės grafinės bylos kodavimas:
  - o Atliekamas skaitmeninio vaizdo kodavimas (suspaudimas) pagal vartotojo nustatytus parametrus;
- ✓ Skaitmeninio grafinio failo dekodavimas:
  - o Atliekamas skaitmeninio vaizdo dekodavimas (atkūrimas) pagal vartotojo nustatytus parametrus;
- ✓ Duomenų (elementų) matricos transformacija, vaizdo spektro pateikimas;
- ✓ Transformuotos duomenų matricos kodavimas;
- ✓ Atvirkštinis transformuotos duomenų matricos dekodavimas:
  - o Atliekamas spektro atkūrimas;
- ✓ Duomenų (elementų) matricos atvirkštinė transformacija;
- ✓ Duomenų išsaugojimas:
  - o Atkurto vaido, vaizdo spektro, užkoduoto vaizdo išsaugojimas (BMP, JPEG, GIF, PNG, TIFF ar ezw formatu);
- ✓ Sistemos dialogo palaikymas su vartotoju:
  - o Sistema vartotojui išveda pranešimus apie sistemoje susidariusias klaidingas ar kitas informatyvias situacijas;
- ✓ Skaitmeninės duomenų bylos (koduoto, nekoduoto vaizdo) užkrovimas;
- ✓ Kodavimo / dekodavimo parametrų nustatymas;
  - o Nustatomas kodavimo slenkstis, kodavimo iteracijų skaičius;
  - o Spektro koeficientų filtravimas papildomais koeficientais.

4.4.2 Nefunkciniai reikalavimai

Sistemai keliami tokie nefunkciniai reikalavimai:

- ✓ Reikalavimai sistemos išvaizdai:
  - o Vartotojo sąsaja paprasta, aiški, neperkrauta elementais. Lengvai suprantamas meniu, intuityvi veiksmų seka;
- ✓ Reikalavimai panaudojamumui:
  - o Minimalus įvedamos informacijos kiekis;
  - o Informavimas apie klaidas;
- ✓ Reikalavimai vykdymo charakteristikoms:

- Kuriamai sistemai konkretūs greičio reikalavimai nėra keliami. Tačiau algoritmo programinė realizacija turi būti optimizuota, kad vartotojas nebūtų verčiamas laukti;

✓ Reikalavimai veikimo sąlygoms:
- Sistemos diegimas vartotojo akimis turi būti paprastas ir lengvai suprantamas;
- Darbo aplinka turi turėti JAVA programų vykdymo aplinką;

✓ Reikalavimai sistemos priežiūrai:
- Kuriama sistema lengvai papildoma naujomis funkcijomis (komponentais);

✓ Reikalavimai saugumui:
- Duomenys (rezultatai) nėra slaptai laikomi nuo pašalinių asmenų;

✓ Kultūriniai-politiniai reikalavimai:
- Sistemoje nėra politinės agitacijos, rasinės, religinės diskriminacijos, pažeminimų ir kt.

✓ Teisiniai reikalavimai:
- Visos technologijos naudojamos sistemos kūrimui yra laisvai prieinamos ir nemokamos.

5. TYRIMAS

5.1 Kokybės vertinimo procesas

Sistemos kokybės proceso įvertinimui pasitelktas ISO 9126 kokybės modelis.

5.1.1 Tinkamumas

Sukurta sistema atlieka ~ 90 % užsakovo nurodytų reikalavimų. Sistemoje nerealizuota tokios funkcijos kaip atkurto vaizdo fragmento suglaudinimas naudojant EZW algoritmą, interaktyvus atkuriamo vaizdo fragmento išrinkimas (šiuo metu tai padaroma papildomos sąsajos pagalba).

5.1.2 Tikslumas

Sistema pateikia teisingus ir sutartus rezultatus.

5.1.3 Apsauga

Sistema nekaupia ir nedirba su slapta informacija, todėl nėra jokios galimybės uždrausti neautorizuotą priėjimą prie programos ar jos duomenų. Sistema sukurta ne komerciniais tikslais. Sistema platinama nemokamai.

5.1.4 Išmokstamumas

Vadovaujantis sistemos vartotojo dokumentacija, sistema lengvai valdoma, valdymas nereikalauja daug pastangų.

5.1.5 Veikimas

Vadovaujantis sistemos vartotojo dokumentacija, sistemos operacijos lengvai valdomos, valdymas nereikalauja daug pastangų.

5.1.6 Laikinė elgsena

Programos kodas, jo struktūra optimizuoti siekiant spartesnio sistemos veikimo. Sistemos veikimo laikas priklauso nuo apdorojamų duomenų kiekio.

5.1.7 Elgsena resursų atžvilgiu

Suprojektuota ir realizuota sistema nereikalauja iš naudojamo kompiuterio aukšto procesoriaus dažnio ar didelio operatyvios atminties kiekio. Tačiau naujesnės (greitesnės) technologijos paskirtas užduotis atliktų greičiau, nei senos architektūros kompiuteriai. Sistemos efektyvumas didinamas optimizuojant matematinius sistemos komponentus, t.y. ieškant greitesnių HT ir EZW skaičiavimo būdų.

5.1.8 Analizuojamumas

Sistemos moduliai ar komponentai, turintys trūkumų ar defektų, gali būti nesunkiai identifikuojami, kadangi sistema suskurta laikantis objektiškai orientuoto programavimo principų, kas palengvina programos kodo „skaitymą“.

5.1.9 Keičiamumas

Klaidos ištaisymas nors ir nėra sudėtingas procesas, tačiau dažnai netgi nežymios klaidos paieška trunka pakankamai ilgai. Klaidos aptikimo pastangos sumažėja dėka objektiškai orientuotos sistemos struktūros.

5.1.10 Stabilumas

Sistema pasižymi stabiliu veikimu. Pagrindinės nestabilaus sistemos veikimo priežastys yra klaidos, neaptiktos testavimo metu.

5.1.11 Testuojamumas

Testavimas yra atsakingas ir gana sudėtingas procesas, tačiau sukurtos sistemos testavimas pakankamai paprastas, kadangi naudojami matematiniai metodai yra pakankamai aiškūs (aiškiai

aprašyti). Testavimo procesą gali apsunkinti nebent testavimo duomenims naudojant didelius duomenų kiekius (laiko atžvilgiu ilgai trunkantis procesas), tačiau ir toks testavimo būdas yra toleruojamas ir netgi būtinas, kuris gali atitikti stresinį testavimo būdą.

5.1.12 Prisitaikomumas

Sistema sukurta Windows XP/2000 operacinėms sistemoms. Kitose, tiek Windows šeimos, tiek Linux, operacinėse sistemose, sukurtas sistema neveiks.

5.1.13 Įdiegiamumas

Sistemos įdiegimo procedūra reikalauja minimalių pastangų. Įdiegimo procedūrai parengtas sistemos įdiegimo gidas (dokumentacija).

5.2 Vertinimo rezultatai

5.1 lentelėje pateikti kokybės vertinimo kriterijai, įvertinti dešimtabalėje sistemoje (0 – blogai, 10 – puikiai).

5.1 lentelė. Programinės įrangos kokybės kriterijų įvertinimas

Kokybės vertinimo kriterijus	Sistemos įvertis
Tinkamumas	9
Tikslumas	10
Apsauga	10
Išmokstamumas	9
Veikimas	9
Laikinė elgsena	8
Elgsena resursų atžvilgiu	8
Analizuojamumas	9
Keičiamumas	9
Stabilumas	9
Testuojamumas	8
Palaikomumas	7
Testuojamumas	9
Prisitaikomumas	8
Įdiegiamumas	10

5.3 Sistemos tobulinimas

Sistemos tobulinimui siūlomi sprendimai:

✓ HT spartinimas – dėl paprastos HT specifikacijos bei remiantis [1], atlikti HT skaičiavimo praktinį pagreitėjimą ir sulyginti su teoriniais rezultatais;

✓ EZW algoritmo optimizavimas – nulinių medžiu paieškos optimizavimas;

✓ LZW, LRE glaudinimo metodų realizavimas – kadangi naudojant spektro koeficientų daugiklį išauga koduojamo vaizdo apimtis, o spektras koduojamas pasikartojančiomis simbolių sekomis, todėl galima bandyti panaudoti tokius glaudinimo algoritmus kaip LZW ar LRE;

✓ Vartotojo sąsajos tobulinimas – interaktyvus fragmento išrinkimas, kelių lygiu fragmento detalizavimo sąsaja.

6. EKSPERIMENTO REZULTATAI

6.1 Eksperimento turinys

- ✓ Skaitmeninio vaizdo kodavimas ir dekodavimas:
  - o Kodavimo realizacija (preliminarūs rezultatai);
  - o Kodavimo metodų vykdymo laikų matavimas;
  - o Kodavimo operacijos greitinimas (Haaro transformacijos greitinimas, EZW algoritmo greitinimas; matuojami parametrai: laikas, kokybė, duomenų apimtis);
  - o Kodavimo metodų modifikavimas siekiant geresnių rezultatų (diskrečiosios Haaro transformacijos spektro modifikavimas; matuojami parametrai: laikas, kokybė, duomenų apimtis).
- ✓ Lokaliai progresyvus kodavimas:
  - o Lokaliai progresyvaus kodavimo rezultatų parametrų matavimas (laikas, operacijų skaičius).

Eksperimento metu naudota techninė bei programinė įranga:

- ✓ Techninė įranga:
  - o Procesorius – Pentium IV (2,93 GHz);
  - o Operatyvioji atmintis – 1 GB RAM.
- ✓ Programinė įranga:
  - o Operacinė sistema – Microsoft Windows XP;
  - o Programavimo aplinka – JAVA 1,5.
  - o Grafikos programa – IrfanView 3.99;

- 1 – 3 semestro metu sukurta skaitmeninių vaizdų kodavimo programa (magistrinis projektas).

6.2 Skaitmeninio vaizdo kodavimas

6.2.1 Kodavimo realizacija

Naudojama diskrečioji Haaro transformacija, progresyvus EZW kodavimo algoritmas. Rezultatai pateikti 6.1 lentelėje bei 6.1 paveikslėlyje.

6.1 lentelė. HT ir EZW metodų rezultatai

Suspaudimo laipsnis, β (slenkstis)	Vaizdo raiška	Vaizdo dydis, KB	Kodavimo trukmė, s	Dekodavimo trukmė, s	PSNR, dB	δ	Vizualinė vaizdo kokybė
1	256x256	64	–	–	–		Originalas (gera)
16 (t_{min} = 1)	256x256	4,019	0,172	0,125	24,062	15,976	Nepatenkinama
70 (t_{min} = 2)	256x256	0,92	0,094	0,11	22,034	20,177	Bloga
222 (t_{min} = 4)	256x256	0,288	0,062	0,078	19,87	25,886	Bloga

Kaip matome, pagal pateiktus atkurtus vaizdus, šiame darbe tiriamas kodavimo metodas stipriai atsilieka nuo pasaulyje plačiai paplitusio JPEG kodavimo standarto.



(a)

HT, EZW **HF (hiperbolinis filtravimas)** **JPEG**



(b) **(c)** **(d)**

(e) **(f)** **(g)**

(h) **(i)** **(j)**

6.1 pav. Eksperimento rezultatai; (a) originalus vaizdas; (b) HT, EZW, kai $t_{min} = 1$, $\beta = 16$; (c) HF, kai $\beta = 16$; (d) JPEG, kai $\beta = 16$; (e) HT, EZW, kai $t_{min} = 2$, $\beta = 70$; (f) HF, kai $\beta = 70$; (g) JPEG, kai $\beta = 70$; (h) HT, EZW, kai $t_{min} = 4$, $\beta = 222$; (i) HF, kai $\beta = 222$; (j) JPEG, kai $\beta = 88$.

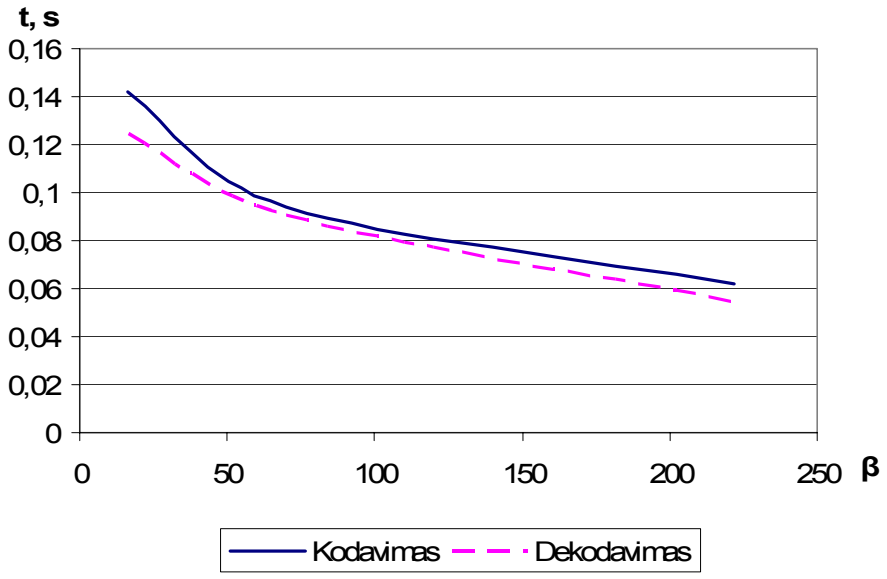
6.2.2 Kodavimo metodų vykdymo laikų matavimas

Kodavimo proceso trukmės – HT + EZW + rezultatų užkrovimas – priklausomybė nuo failo raiškos bei kodavimo slenksčio (spaudimo laipsnio):

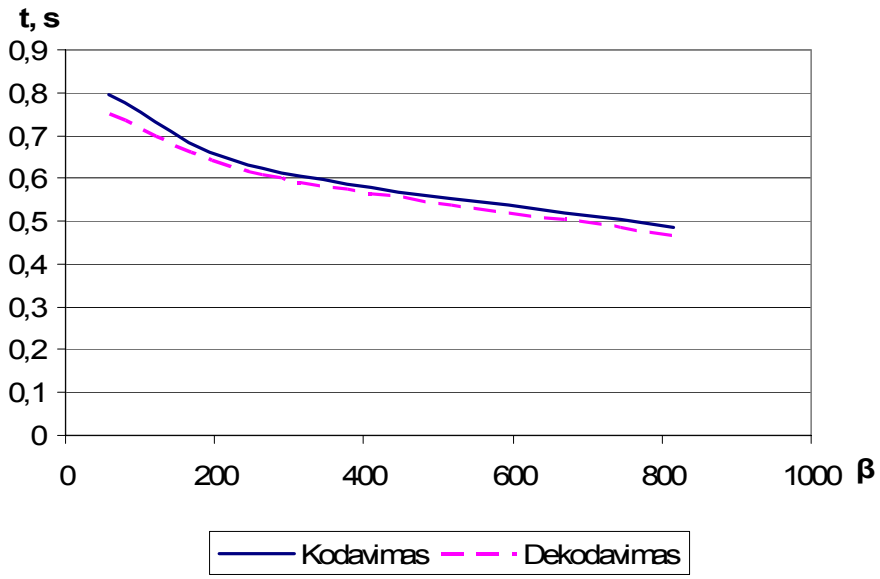
6.2 lentelė. HT ir EZW metodų vykdymo laikų rezultatai

Vaizdo raiška	Slenkstis t_{min} (β)	Vaizdo apimtis, KB	Kodavimas, *s*	Dekodavimas, *s*
256x256	1 (16)	4,019	0,142	0,125
	2 (70)	0,92	0,094	0,091
	4 (222)	0,288	0,062	0,055
512x512	1 (57)	4,494	0,797	0,75
	2 (264)	0,971	0,625	0,609
	4 (815)	0,314	0,485	0,465
1024x1024	1 (214)	4,785	2,797	2,315
	2 (1005)	1,018	2,254	1,819
	4 (3413)	0,3	1,925	1,397

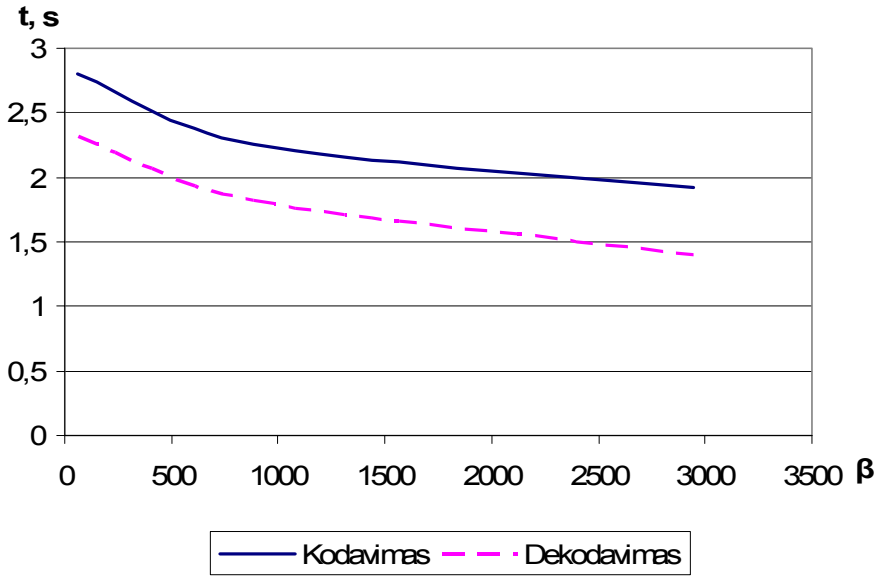
Iš 6.2–oje lentelėje pateiktų kodavimo proceso trukmės duomenų, galima pastebėti tokią tendenciją – užkoduotų vaizdų duomenų apimtis prie kiekvieno kodavimo slenksčio nepriklausomai nuo vaizdo raiškos yra panašios. Matome, kad tiek 256x256, tiek 512x512, tiek 1024x1024 užkoduoti vaizdai (kai t_{min} = 1) užima apie ~4–5 KB atminties, nors tokios raiškos vaizdai atitinkamai yra 64 KB, 256 KB ir 1024 KB atminties. Nekoduotų vaizdų apimtis didėja po 4 kartus, užkoduotų vaizdų apimtis išlieka beveik vienoda. Jei atkreipsime dėmesį į suspaudimo laipsnį β, pastebėsime lygia tą pačia tendencija, kuomet koduojamų vaizdų apimtis didėja po 4 kartus, suspaudimo laipsnis taip pat didėja ~4 kartus esant tam pačiam kodavimo slenksčiui t_{min}. Iš to galima spręsti, kad nepriklausomai nuo koduojamo vaizdo raiškos, suspaustas vaizdas yra atkuriamas iš panašios koeficientų aibės (kiekybės prasme).



6.2 pav. Vaizdo 256x256 kodavimo ir dekodavimo operacijų trukmės grafikas pagal suspaudimo laipsnį

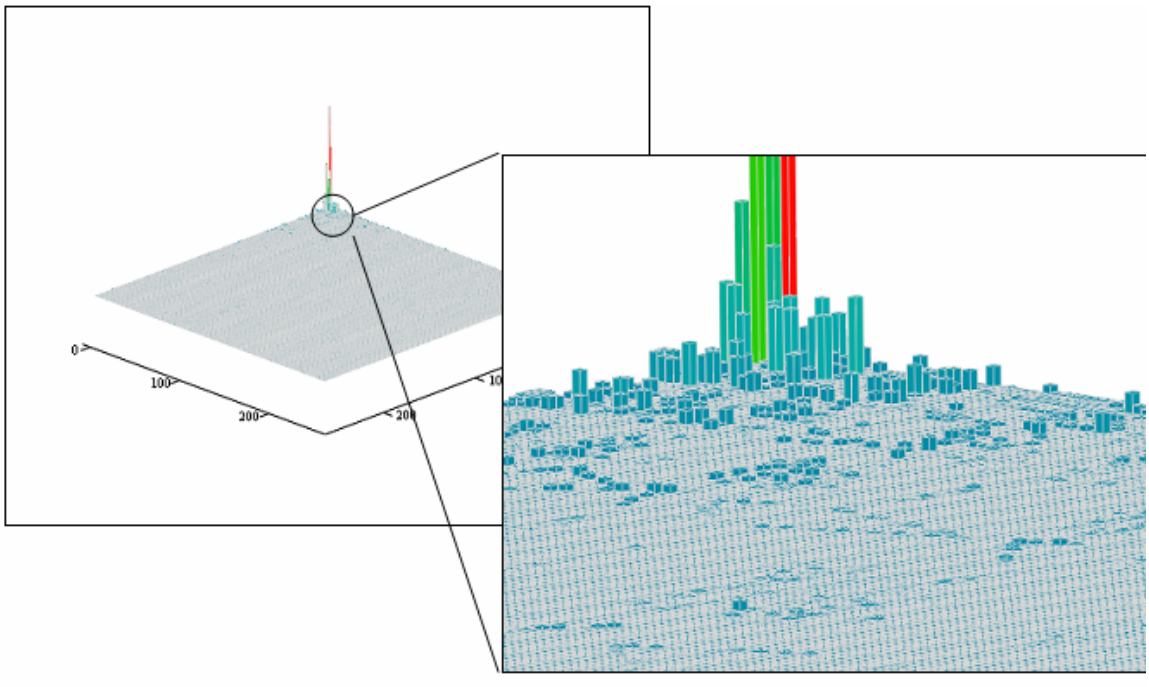


6.3 pav. Vaizdo 512x512 kodavimo ir dekodavimo operacijų trukmės grafikas pagal suspaudimo laipsnį

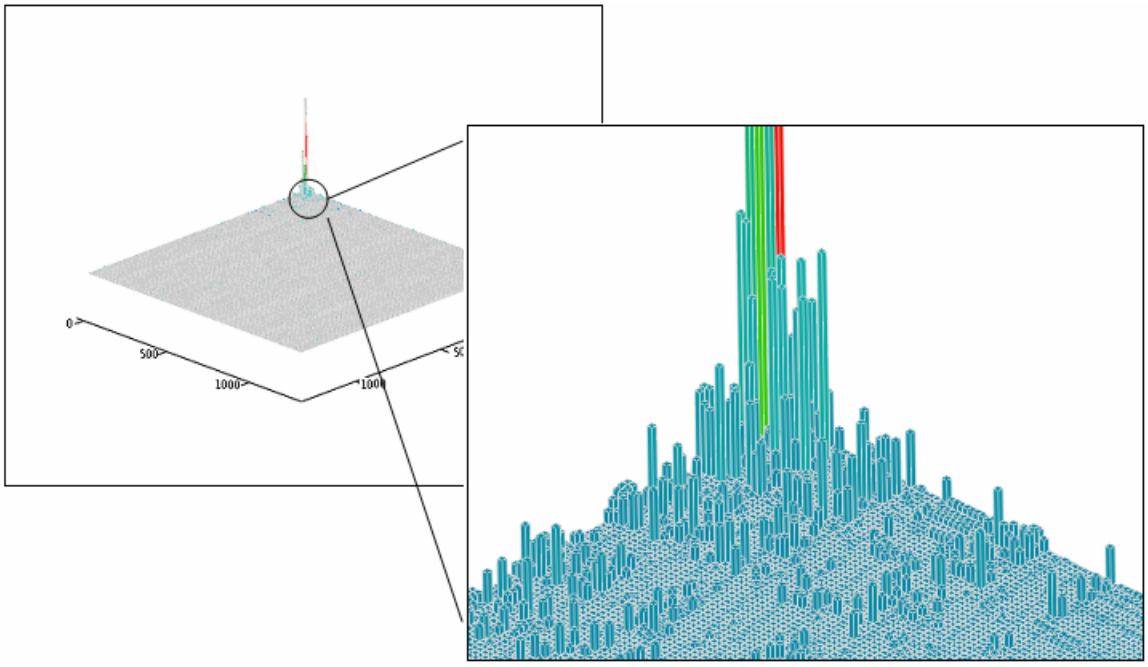


6.4 pav. Vaizdo 1024x1024 kodavimo ir dekodavimo operacijų trukmės grafikas pagal suspaudimo laipsnį

Iš 6.5 paveikslėlio matome, jog reikšmingų spektro koeficientų aibė yra panaši tiek 256x256, tiek 1024x1024 vaizdams. Todėl natūralu, jog atkurto vaizdo kokybė prastėja didėjant jo raiškai.



(a)



(b)`

6.5 pav. Vaizdo spektras ir reikšmingi spektro koeficientai; (a) 256x256 raškos vaidas; (b) 1024x1024 raiškos vaizdas

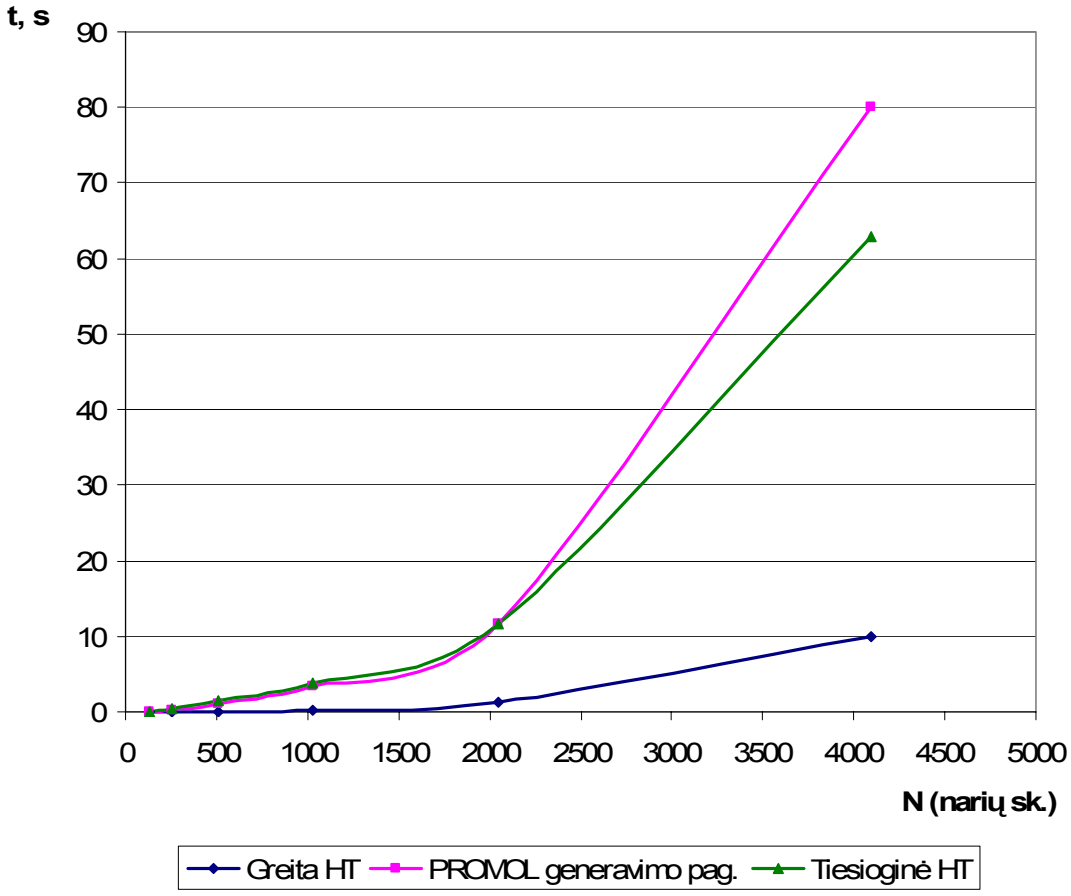
6.2.3 Kodavimo operacijos spartinimas

Siekiant spartesnio sistemos darbo, kodavimui naudojami metodai turi būti suprogramuoti optimaliai. Ieškant greitesnio diskrečiosios Haaro transformacijos apskaičiavimo metodo, buvo

atlikti HT spektro skaičiavimų trukmės matavimai programuojant skirtingais programavimo metodais:

- ✓ HT apdorojama tiesioginiu būdu naudojant JAVA programavimo kalbą;
- ✓ HT apskaičiuojama greituoju būdu naudojant JAVA programavimo kalbą [1];
- ✓ Open PROMOL kalba sugeneruoti HT komponentai (JAVA kalbai).

Šių metodų HT apdorojimo laikai žymiai skiriasi esant dideliems vaizdams (pvz., 1024x1024 ir didesni). Tai akivaizdžiai matosi 6.6–ame pav., kur pateiktos HT skaičiavimo trukmės naudojant paminėtus metodus. Dėka paprastos HT skaičiavimo specifikos, realizuotą HT procedūrą, pagal [1], galima nežymiai paspartinti. Tokiu atveju aritmetiniams sumų ir skirtumų vidurkiams naudojamas vektoriaus narių poras galime pakeisti didesnėmis vektoriaus narių aibėmis (t.y. naudojamos narių porų poros, poros porų poros ir t.t.).



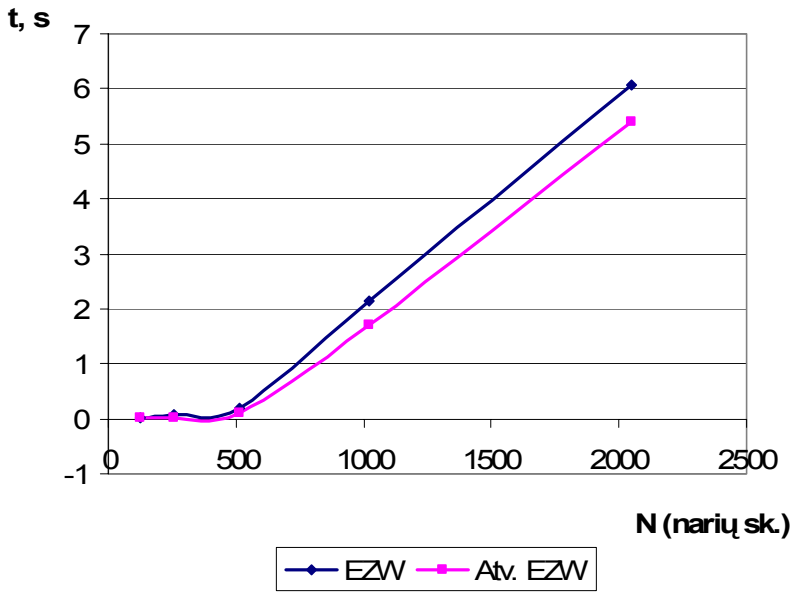
6.6 pav. HT skaičiavimo trukmės (greita HT, PROMOL kalbos komponentais grįsta HT, tiesioginė HT)

Kalbant apie EZW kodavimo metodą, galima išskirti dvi algoritmo optimizavimo sritis:

- ✓ EZW kodavimo algoritmas (spektro užkodavimui);

- ✓ Atvirkštinis EZW algoritmas (spektro atkūrimui).

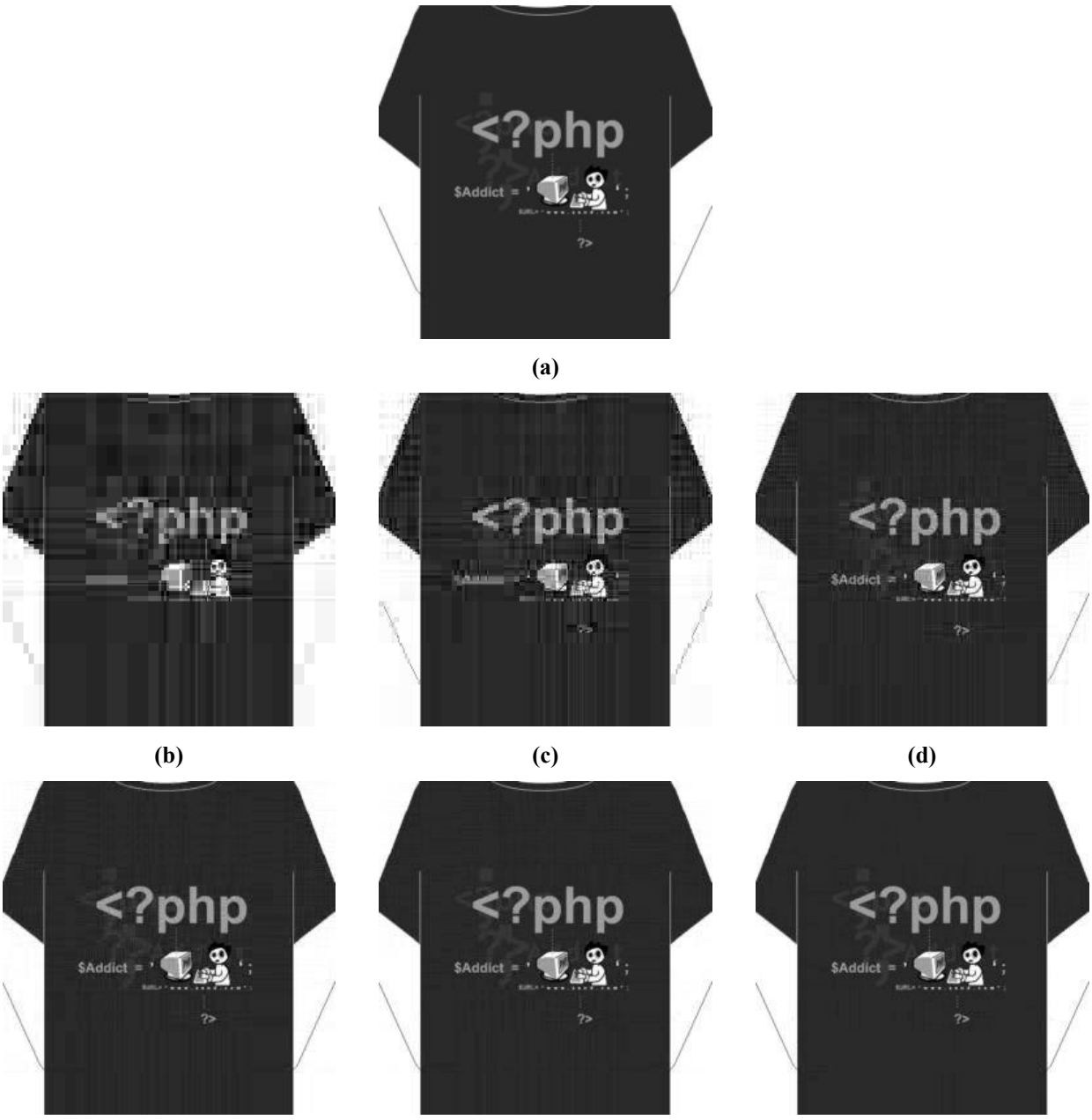
Pirmuoju atveju aktualu mokslininkų vis dar ieškoma greita/optimali nulinių medžių paieška diskrečiame Haaro spektre. Tai vienas iš didžiausių šio naujo progresyvaus kodavimo algoritmo trūkumų. Antruoju atveju, atvirkštinis EZW algoritmas dėkingesnis tuo, kad jo metu nėra atliekama jokia nulinių medžių paieška. Tiesioginio ir atvirkštinio EZW algoritmo vykdymo greičių priklausomybė nuo spektro dydžio pateikta 6.7–ame paveikslėlyje.



6.7 pav. Tiesioginio ir atvirkštinio EZW algoritmo vykdymo greičių priklausomybė nuo spektro dydžio (*N*x*N*)

6.2.4 Tikslinis kodavimo metodų modifikavimas

Kadangi, modifikavus spektrą papildomu koeficientu, užkoduoti duomenys pakankamai išsiplečia (gerėja atkuriamo vaizdo kokybė) ir kodavimas atliekamas 6 specialiais simboliais, galima atlikti papildomą kodavimą taikant informacijos neprarandančius kodavimo algoritmus. Šiame eksperimente taikomi anksčiau aprašyti RLE ir LZW (RLE arba LZW, bet ne abu kartu). Ir, vaizdo kokybės bei apimties atžvilgiu, pasiekiami pakankamai geri glaudinimo rezultatai. Spektro (256x256) daugiklio įtakos užkoduoto vaizdo apimčiai, papildomų glaudinimo algoritmų eksperimentiniai rezultatai pateikti 6.8 paveikslėlyje ir 6.3 lentelėje



(a)

(b) (c) (d)

(e) (f) (g)

6.8 pav. Eksperimento rezultatai: spektro (256x256) reikšmių kėlimo daugikliu *c* įtaką atkuriamam vaizdui; (a) originalas; (b) *c* = 1, PSNR = 24,062 dB (sistema nemodifikuota); (c) *c* = 2, PSNR = 26,82 dB; (d) *c* = 5, PSNR = 32,095 dB; (e) *c* = 10, PSNR = 36,308 dB; (f) *c* = 20, PSNR = 39,084 dB; (g) *c* = 50, PSNR = 39,323 dB.

6.3 lentelė. Spektro daugiklio įtaka koduojamo vaizdo apimčiai

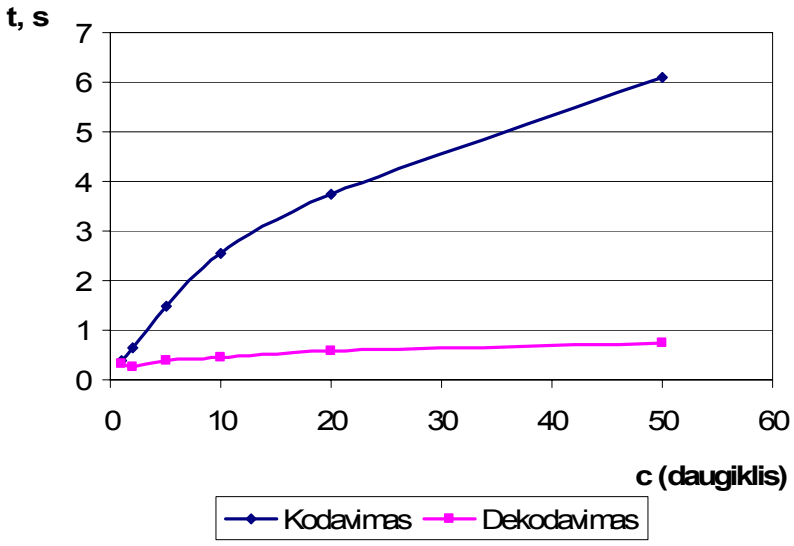
	Spektro daugiklis *c*					
	1	**2**	**5**	**10**	**20**	**50**
Apimtis, KB	4,019	9,249	20,828	33,264	48,649	74,063
PSNR, dB	24,062	26,82	32,095	36,308	39,084	39,323
δ	15,976	11,623	6,336	3,901	2,834	2,757
Apimtis, KB (LRE)	2,732	7,214	14,58	24,283	36,487	59,25
Apimtis, KB (LZW)	1,122	2,152	4,143	6,367	9,413	14,567

Po EZW kodavimo atlikto LRE glaudinimo efektyvumas < 2, o LZW atveju efektyvumas siekė net iki 5.

6.4 lentelė. Spektro daugiklio įtaka kodavimo trukmei

	Spektro daugiklis *c*					
	1	**2**	**5**	**10**	**20**	**50**
Kodavimas, *s*	0,375	0,64	1,469	2,546	3,75	6,094
Dekodavimas, *s*	0,313	0,25	0,391	0,454	0,594	0,75
Kodavimas LRE, *s*	0,021	0,055	0,11	0,17	0,32	0,53
Kodavimas LZW, *s*	0,002	0,005	0,008	0,015	0,031	0,045

Reikia pastebėti, jog palyginus LRE ir LZW, pastarasis turi ne tik geresnį glaudinimo koeficientas, tačiau pati procedūra trunka ~10 kartų trumpiau ir labai nežymiai prisideda prie bendros vaizdo glaudinimo trukmės.



6.9 pav. Spektro daugiklio įtaka vaizdo kodavimo ir dekodavimo trukmei

Daugiau eksperimento rezultatų pateikta priede.

6.3 HT spektro radimo procedūros teorinis efektyvumo įvertinimas

6.3.1 Vienmatis atvejis

Paprasta diskrečiųjų Haaro transformacijų spektrų radimo procedūrų analizė leidžia nesunkiai įvertinti laiko sąnaudas. Taip pat galime atlikti lyginamąją analizę tarp tiesioginės spektro radimo procedūros ir procedūros aprašytos 2.5.1 skyrelyje (diskrečiojo Haaro spektro apskaičiavimo vaizdo fragmentams algoritmas; vienmatis atvejis). Nagrinėkime diskrečiąją Harro

transformaciją vaizdo fragmentui X_k, susietu su spektro koeficientu $Y(k)$. Naudojant tiesioginę Haaro transformaciją operacijų (sudėties, daugybos) skaičių galime apskaičiuoti taip:

$$\Re_t = 2^s + \sum_{t=0}^{s-1} 2^{s-t}(s-t). \quad \textbf{(6.1)}$$

Tuo tarpu netiesioginės procedūros, apteiktos 2.5.1 skyrelyje, operacijų skaičius yra randamas pagal formulę:

$$\Re_n = 2^s + 2(n-s) - 1. \quad \textbf{(6.2)}$$

Iš operacijų skaičiaus ir jų santykio, pateikto 6.5 lentelėje, aiškiai matome, kad netiesioginė procedūra yra daug efektyvesnė laiko sąnaudų atžvilgiu.

6.5 lentelė. Tiesioginės ir netiesioginės procedūrų operacijų skaičius, jų palyginimas, kai $n = 11$

Parametras *s*	**5**	**6**	**7**	**8**	**9**	**10**	**11**
$\Re_t$	290	706	1666	3842	8706	19548	43010
$\Re_n$	45	75	137	263	517	1027	2049
$\Re_t / \Re_n$	6,444	9,413	12,161	14,608	16,839	18,946	20,991

Iš 6.5 lentelės matome, kad operacijų skaičiaus augimas, didėjant *s*, netiesioginės procedūros atveju yra mažesnis negu tiesioginės procedūros atveju. Taigi netiesioginė procedūra teoriškai turėtų būti dar efektyvesnė esant dideliems *s* vertėms.

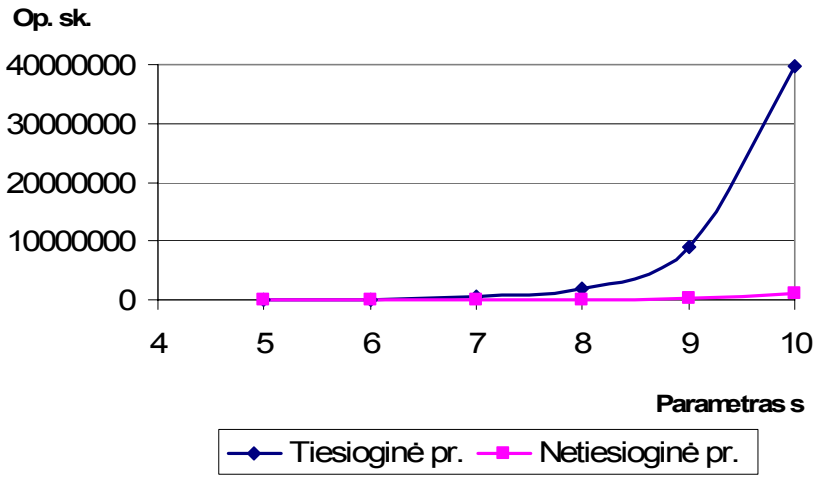
6.3.2 Dvimatis atvejis

Dvimačiu atveju, 6.3.1 skyrelyje pateikti operacijų skaičiaus įverčiai, kai nagrinėjamas vaizdas yra kvadratinis ir jo dydis yra $2^s \times 2^s$, keičiami žemiau pateiktais įverčiais. Tiesioginei procedūrai:

$$\Re_t = 2^{s+1} + (2^s + \sum_{t=0}^{s-1} 2^{s-t}(s-t)), \quad \textbf{(6.3)}$$

netiesioginiai procedūrai:

$$\Re_n = (2^s - 1)^2 + 4(n-s)(2^s - 1) + 10(n-s) - 4. \quad \textbf{(6.4)}$$



6.10 pav. Tiesioginės ir netiesioginės procedūrų operacijų palyginimas, kai $n = 11$

6.6 lentelė. Tiesioginės ir netiesioginės procedūrų operacijų skaičius, jų palyginimas, kai $n = 11$

Parametras *s*	**5**	**6**	**7**	**8**	**9**	**10**
$\Re_t$	18560	90,68	426496	1967104	8914944	39849984
$\Re_n$	1761	5275	18197	68111	265225	1050627
$\Re_t / \Re_n$	10,539	17,131	23,428	28,881	33,613	37,93

Iš 6.6 lentelės ir 6.10 paveikslėlio matome, kad kaip ir vienmačiu atveju, netiesioginis spektro radimas yra daug efektyvesnis operacijų skaičiaus prasme.

6.4 HT spektro radimo procedūros praktinis efektyvumo įvertinimas

6.4.1 HT spektro radimo laiko sąnaudų priklausomybė nuo fragmento vietos vaizde

Programos pagalba patikrinsime, ar procedūros realizacijos vykdymo laikas priklauso nuo fragmento lokalizacijos vaizde. Tam tikslui rasime 8–ių skirtingų 256x256 dydžio fragmentų spektrus ir patikrinsime, per kiek laiko jie buvo apskaičiuoti, taikant skirtingus spektrų radimo metodus.

6.7 lentelė. Diskretaus Haaro spektro radimo procedūrų laiko sąnaudų įverčių priklausomybė nuo fragmento vietos vaizde

Fragmentas	$X_{8,8}$	$X_{8,11}$	$X_{11,8}$	$X_{12,12}$	$X_{10,13}$	$X_{15,12}$	$X_{8,15}$	$X_{15,15}$
$\Re_t$, s	34,84	34,81	34,81	34,88	34,79	34,94	34,71	34,81
$\Re_n$, s	2,81	2,82	2,84	2,83	2,81	2,81	2,8	2,86
$\Re_t / \Re_n$	12,399	12,344	12,257	12,325	12,381	12,434	12,396	12,171

Pastaba: $\Re_t$ (tiesioginis algoritmas) ir $\Re_n$ (netiesioginis algoritmas) procedūros atliktos po 200 kartų.

Pagal 6.7 lentelės įverčius galime daryti išvadą, kad procedūrų realizacijos laikas nepriklauso nuo fragmento lokalizacijos vaizde. Fragmento radimo laikas taikant netiesioginę procedūrą yra daugiau nei 10 kartų trumpesnis negu taikant tiesioginę procedūrą.

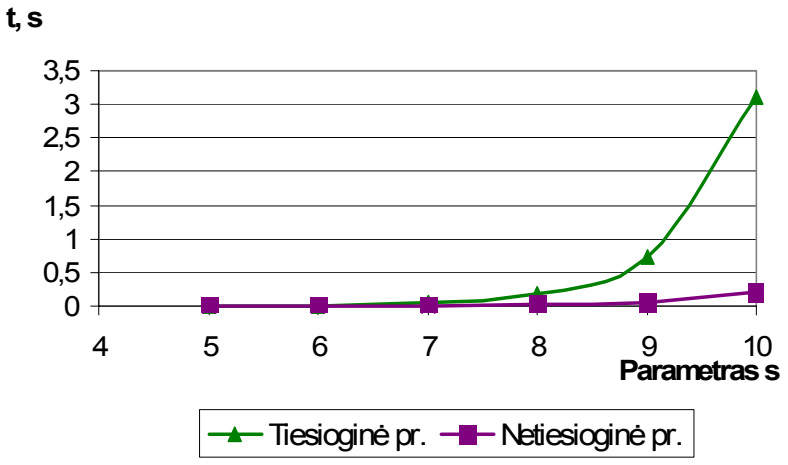
6.4.2 HT spektro radimo laiko sąnaudų priklausomybė nuo fragmento dydžio

Tai pat atliksime eksperimentą, kuris randa diskretaus Haaro spektro apskaičiavimo laiko įvertį, priklausantį nuo fragmento dydžio. Kadangi, kaip jau įsitikinome, fragmento lokalizacija vaizde spektro radimo laikui įtakos neturi, eksperimento metu ieškosime spektro vaizdo fragmentams, susietiems su spektro koeficientais $Y(2^{n-s}, 2^{n-s})$.

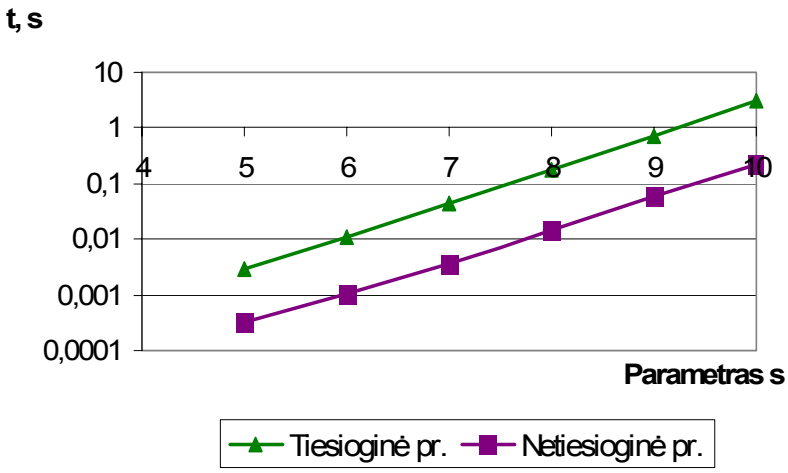
6.8 lentelė. Diskretaus Haaro spektro radimo procedūrų laiko sąnaudų įverčių priklausomybė nuo fragmento dydžio

Parametras *s*	**5**	**6**	**7**	**8**	**9**	**10**
$\Re_t, s$	2,86	11,16	43,78	175,13	726,11	3118,75
$\Re_n, s$	0,33	1,03	3,67	14,22	58,77	218,63
$\Re_t / \Re_n$	8,667	10,835	11,93	12,32	12,36	14,26

Pastaba: $\Re_t$ (tiesioginis algoritmas) ir $\Re_n$ (netiesioginis algoritmas) procedūros atliktos po 1000 kartų.



6.11 pav. Tiesioginės ir netiesioginės procedūrų laiko sąnaudų įverčių priklausomybė nuo fragmento dydžio



6.12 pav. Tiesioginės ir netiesioginės procedūrų laiko sąnaudų įverčių priklausomybė nuo fragmento dydžio (logaritminė skalė)

Šiuos rezultatus galime palyginti su teoriniais rezultatais (6.6 lentelė). Matome, kad esant mažam parametrui s, $\Re_t / \Re_n$ beveik sutampa su teoriniais rezultatais, tačiau didėjant s, laiko santykis nėra toks didelis, kaip buvo galima tikėtis. Taip galėjo atsitikti dėl programavimo stiliaus, naudojamos platformos ir pan. Kartais įgyvendinant algoritmą tenka rinktis ilgesnį, ne tokį efektyvų kelią. Taip pat panašu, kad operacijų padidėjimas priklauso nuo parametro s eksponentiškai, nes mažiems s eksperimento rezultatai beveik sutampa su prognoze, o didėjant parametrui s, jie vis labiau skiriasi nuo prognozuotų rezultatų.

6.4.3 HT spektro radimo laiko sąnaudų priklausomybė nuo vaizdo dydžio

Atliksime eksperimentą, kuris parodys abiejų procedūrų efektyvumą, keičiantis viso vaizdo dydžiui. Eksperimentą atliksime, kai s = 5, 6, 7, o n = 8, 10.

6.9 lentelė. Diskretaus Haaro spektro radimo procedūrų laiko sąnaudų įverčių priklausomybė nuo vaizdo dydžio

Parametrai *s*, *n*	**5, 8**	**6,8**	**7,8**	**5,10**	**6,10**	**7,10**
$\Re_t$, s	5,67	21,92	86,27	5,67	22,03	86,56
$\Re_n$, s	0,58	1,92	6,98	0,63	2	7,19
$\Re_t / \Re_n$	9,776	11,417	12,36	9	11,015	13,039

Pastaba: $\Re_t$ (tiesioginis algoritmas) ir $\Re_n$ (netiesioginis algoritmas) procedūros atliktos po 2000 kartų.

Kaip matome iš eksperimentų (6.9 lentelė), didėjant parametrui *n*, optimizuoto algoritmo vykdymo laikas nežymiai išauga (~ 3 – 8 %), tačiau tiesioginio algoritmo metu jis nekinta (pakinta < 0,5 %), nes tiesioginio algoritmo operacijų skaičius priklauso tik nuo fragmento dydžio. Iš to išplaukia, kad netiesioginio metodo santykinis efektyvumas mažėja, kai didėja viso vaizdo dydis.

7. IŠVADOS

- ✓ Nespalvotų vaizdų praradiminio kodavimo (suglaudinimo) algoritmų klasėje geriausi apdorojimo rezultatai priskirtini tiems algoritmams, kurie „dirba“ spektrinėje srityje ir tikslinai išnaudoja specifinius diskrečiųjų bangelių transformacijų savybes. Tai JPEG2000 standartas, EZW algoritmas, SPIHT (Set Partitioning in Hierarchical Trees) algoritmas ir pan.
- ✓ Darbe analizuojama nauja lokaliai progresyvaus nespalvotų vaizdų kodavimo idėja (metodas). Metodas grindžiamas greitu perėjimu nuo diskrečiojo Haaro (HT) bangelių spektro visam vaizdui prie HT spektro pasirinktiems vaizdo fragmentams bei progresyviojo vaizdų kodavimo algoritmu EZW.
- ✓ Diskrečioji Haaro transformacija yra gana paprasta, pasižymi geromis lokalizavimo pagal dažnį ir erdvėje savybėmis, turi greitus skaičiavimo algoritmus, tačiau dėl prasto informacijos „pakavimo“ spektrinėje srityje (lėtai gęsta spektrinės komponentės) jos panaudojimas progresyvaus vaizdų kodavimo srityje (taikant EZW algoritmą) yra problematiškas.
- ✓ Tikslinga būtų realizuoti darbe siūlomą idėją, taikant sudėtingesnes ir kur kas geresnėmis („pakavimo“, konvergavimo) savybėmis pasižyminčias bangeles (Le Gall, Daubechie ir pan.) ir jų bazėje sukonstruotas DBT.
- ✓ Šiame darbe siekiant geresnių rezultatų buvo atliktos kelios kodavimo mechanizmo modifikacijos: spektro koeficientų modifikavimas (daugyba iš papildomo koeficiento) bei įnešti nepraradiminiai vaizdų kodavimo (po EZW) algoritmai (RLE, LZW).

LITERATŪRA

[1] **Chang P.; Piau P. „**Modified Fast and Exact Algorithm for Fast Haar Transform“. *International Journal of Computer Science and Engineering*, 2008, *Vol*. 2, *No*. 2, 55–58.

[2] **Holtz K.**, „Image and video compression techniques“ [Žiūrėta 2006 11 11], prieiga internete <http://www.autosophy.com/videcomp.htm>

[3] **Laurens L**. „RLE compression“. [Žiūrėta 2008.03.14] prieiga internete <http://www.prepressure.com/library/compression_algorithms/rle>

[4] **Marshall D.** „Lempel-Ziv-Welch (LZW) algorithm“.[Žiūrėta 2008.03.15] prieiga internete <http://www.cs.cf.ac.uk/Dave/Multimedia/node214.html>

[5] **Matthew R.** „JPG or lossless for image archives“.[Žiūrėta 2006 10 28], prieiga internete <http://www.wfu.edu/~matthews/misc/jpg_vs_gif/JpgCompTest/JpgForArchive.html>

[6] **Netherlands S.** „Wavelet Image Compressor — MinImage“ [Žiūrėta 2006.11.25], prieiga internete <http://www.springerlink.com/content/x5v63t7971274245/>

[7] **Shapiro J.M.** „Embedded Image Coding Using Zero-trees of Wavelet Coefficients“. IEEE Transactions on Signal Processing, 1993, *Vol*. 41, *No*. 12, 3445–3462.

[8] **Valantinas J.** „On the application of Haar wavelet to locally progressive encoding of grey-level images“. *ISSN 1392 – 124X Information technology and control,* 2007, *Vol*. 36, *No*. 2, 177–186.

[9] **Valantinas J.** „Skaitmeninių vaizdų efektyvaus kodavimo, analizės bei sintezės metodų sudarymas ir tyrimas“. Kaunas. 2006.

[10] **Xiao P.** „Image Compression By Wavelet Transform“, Master Thesis, East Tennessee State University, 2001.08.

[11] **Zhao Y.** „Embedded image coding algorithm using wavelet and vector quantization“. Master of Science. Department of Electrical and Computer Engineering and Computer Science of the college of Engineering, B.S.Shanghai Jiaotong University. China. 2004.

[12] **Žemlo G. „**Fraktalinio suspaudimo metodo tyrimas“. Magistro darbas. VPU Matematikos ir Informatikos fakultetas, 2004.

SUMMARY

Digital images are widely used in computer applications. Uncompressed digital images require considerable storage capacity and transmission bandwidth. Efficient image compression solutions are becoming more critical with the recent growth of data intensive, multimedia-based web applications.

In this paper, a novel locally progressive image encoding idea (procedure) is presented. The procedure explores both specific properties of Haar wavelets and the EZW algorithm originally used for progressive image encoding. Properties of various discrete transforms and areas of their practical applicability are discussed too. Preliminary experimental results show that the joint application of the discrete Haar transform and the EZW algorithm to locally progressive compression of digital images is not effective. To increase efficiency of the approach (idea), some modifications are proposed. In particular, appropriately chosen enlargement of Haar spectral coefficients led to much better overall performance, unfortunately, at the expense of time expenditures.

SANTRUMPŲ IR TERMINŲ ŽODYNĖLIS

2D – dvimatis vaizdas;

3D – trimatis vaizdas;

DBT – diskrečioji bangelių transformacija;

DFT – diskrečioji Furjė transformacija;

DKT – diskrečioji kosinusinė transformacija;

DT – diskrečioji transformacija;

ESM – elektroninė skaičiavimo mašina;

EZW – kodavimo algoritmas (Embedded Zero-tree Wavelet);

GFT – greitoji Furjė transformacija;

HDTV – skaitmeninės televizijos atmaina, kurioje televiziniai (TV) signalai perduodami naudojant skaitmeninius kodavimo ir spūdos metodus (High Definition television);

HF – hiperbolinis filtravimas;

HT – Haaro transformacija;

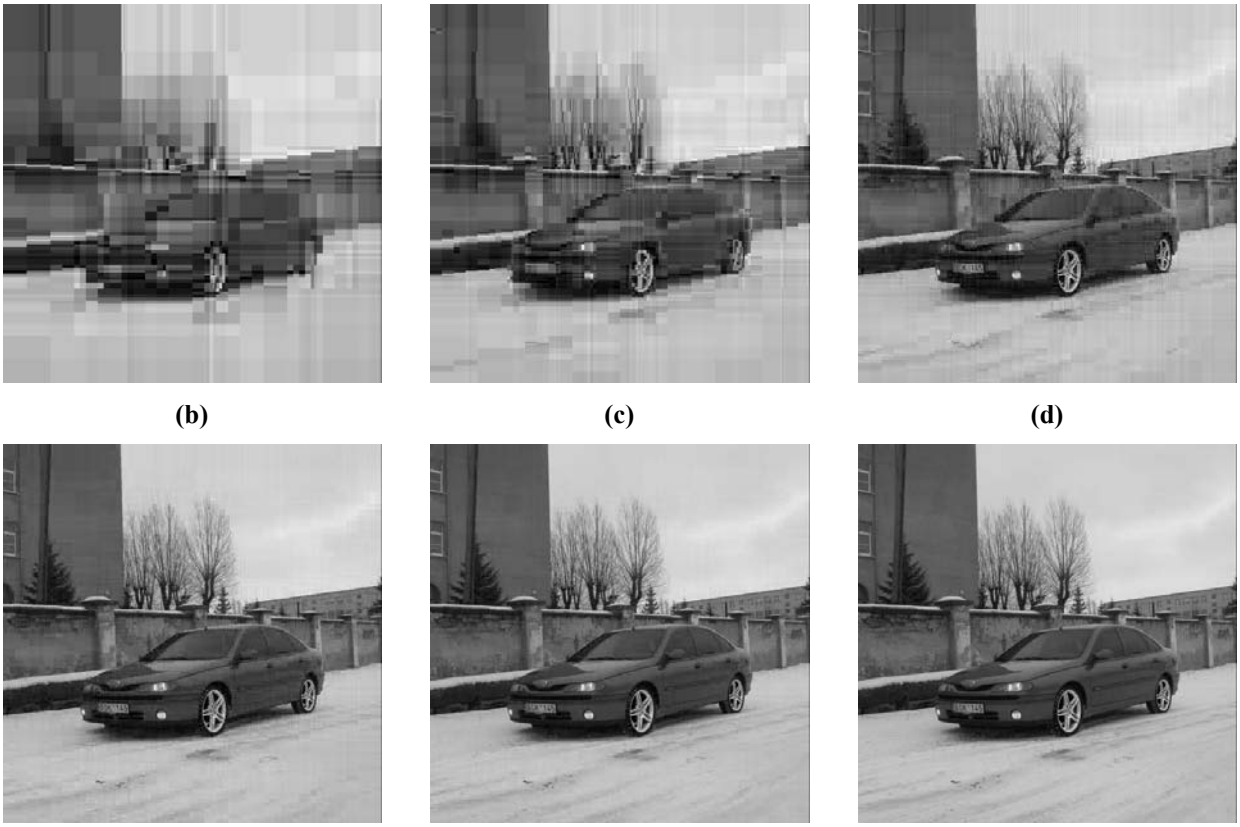
PSNR – didžiausias santykis signalas/triukšmas (Peak Signal-to-Noise Ratio).

PRIEDAI

1. PRIEDAS. TIKSLINIO KODAVIMO METODŲ MODIFIKAVIMO REZULTATAI 512X512 VAIZDAMS



(a)



(b) **(c)** **(d)**

(e) **(f)** **(g)**

1.1 pav. Eksperimento rezultatai: spektro (512x512) reikšmių kėlimo daugikliu *c* įtaką atkuriamam vaizdui; (a) originalas „*laguna.bmp*“, 256 KB; (b) *c* = 1, PSNR = 24,321 dB (sistema nemodifikuota); (c) *c* = 2, PSNR = 26,342 dB; (d) *c* = 5, PSNR = 29,649 dB; (e) *c* = 10, PSNR = 32,681 dB; (f) *c* = 20, PSNR = 36,634 dB; (g) *c* = 50, PSNR = 41,536 dB.

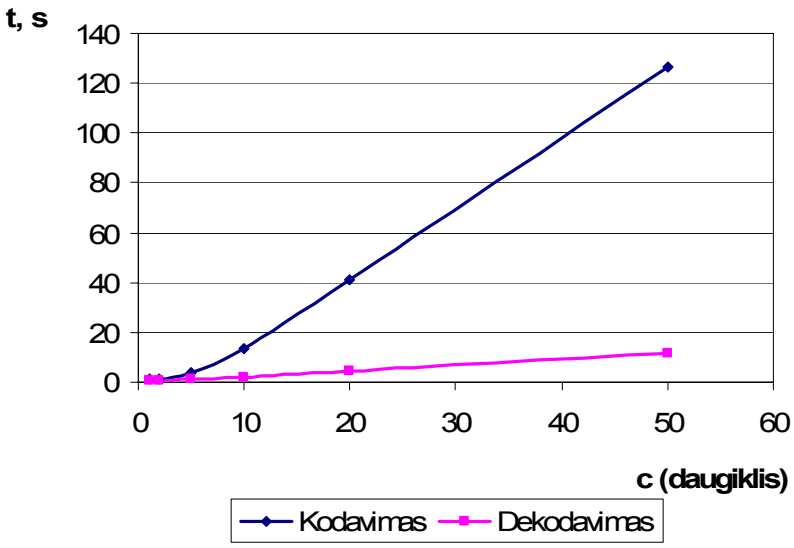
1.1 lentelė. Spektro daugiklio įtaka koduojamo vaizdo apimčiai

	Spektro daugiklis *c*					
	1	**2**	**5**	**10**	**20**	**50**
Apimtis, KB	3,058	8,445	28,035	62,057	120,052	249,441
PSNR, dB	24,321	26,342	29,649	32,681	36,634	41,536
δ	15,506	12,287	8,397	5,923	3,757	2,137
Apimtis, KB (LRE)	1,975	5,464	19,188	44,474	90,51	196,324
Apimtis, KB (LZW)	0,854	1,965	5,577	11,879	23,229	49,061

Eksperimentui naudotas vaizdas „*laguna.bmp*“ suspaustas naudojant JPEG standartą užima 38,6 KB atminties.Spektro (512x512) daugiklio įtaka vaizdo kodavimo ir dekodavimo trukmei:

1.2 lentelė. Spektro daugiklio įtaka kodavimo trukmei

	Spektro daugiklis *c*					
	1	**2**	**5**	**10**	**20**	**50**
Kodavimas, *s*	1,062	1,39	3,938	13,265	41,344	126,414
Dekodavimas, *s*	0,906	0,813	1,062	1,954	4,422	11,531
Kodavimas LRE, *s*	0,03	0,04	0,15	0,34	0,67	1,42
Kodavimas LZW, *s*	0,003	0,008	0,015	0,031	0,046	0,078



1.2 pav. Spektro daugiklio įtaka vaizdo kodavimo ir dekodavimo trukmei

2. PRIEDAS. TIKSLINIO KODAVIMO METODŲ MODIFIKAVIMO REZULTATAI 1024X1024 VAIZDAMS



(a)



(b)



(c)



(d)



(e)



(f)



(g)

2.1 pav. Eksperimento rezultatai: spektro (1024x1024) reikšmių kėlimo daugikliu *c* įtaką atkuriamam vaizdui; (a) originalas „*tower.bmp*“, 1 MB; (b) *c* = 1, PSNR = 22,984 dB (sistema nemodifikuota); (c) *c* = 2, PSNR = 24,863 dB; (d) *c* = 5, PSNR = 27,948 dB; (e) *c* = 10, PSNR = 30,542 dB; (f) *c* = 20, PSNR = 33,587 dB; (g) *c* = 50, PSNR = 38,751 dB.

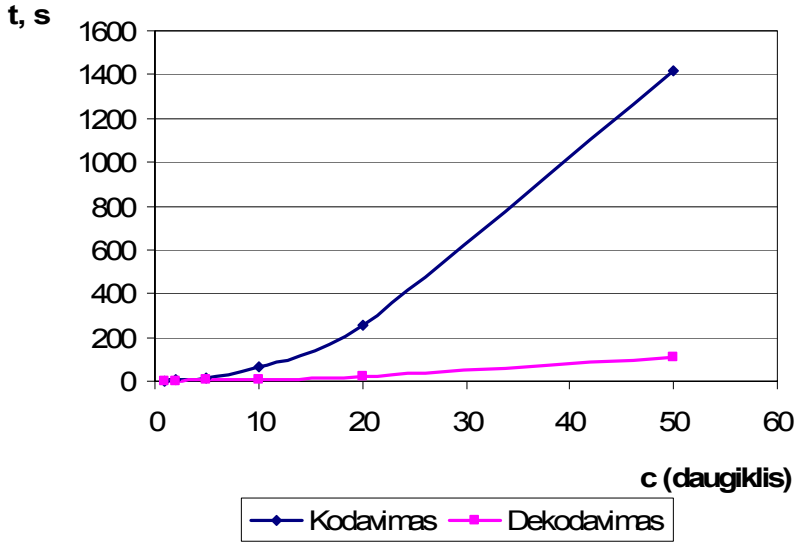
2.1 lentelė. Spektro daugiklio įtaka koduojamo vaizdo apimčiai

	Spektro daugiklis *c*					
	1	**2**	**5**	**10**	**20**	**50**
Apimtis, KB	4,794	15,781	59,65	141,896	301,661	678,587
PSNR, dB	22,984	24,863	27,948	30,542	33,587	38,751
δ	18,087	14,568	10,213	7,576	5,336	2,944
Apimtis, KB (LRE)	2,985	9,773	38,443	95,184	212,372	523,839
Apimtis, KB (LZW)	1,2	3,146	10,449	24,397	53,132	130,487

Eksperimentui naudotas vaizdas „*tower.bmp*“ suspaustas naudojant JPEG standartą užima 167,96 KB atminties.

2.2 lentelė. Spektro daugiklio įtaka kodavimo trukmei

	Spektro daugiklis *c*					
	1	**2**	**5**	**10**	**20**	**50**
Kodavimas, *s*	2,515	3,797	14,563	62,64	257,875	1417,296
Dekodavimas, *s*	2,406	2,75	4,219	8,36	24,578	107,938
Kodavimas LRE, *s*	0,03	0,09	0,31	0,75	1,64	3,79
Kodavimas LZW, *s*	0,007	0,015	0,031	0,047	0,078	0,125



2.2 pav. Spektro daugiklio įtaka vaizdo kodavimo ir dekodavimo trukmei

3. PRIEDAS. APIE LOKALIAI PROGRESYVŲJĮ DVIMAČIŲ SKAITMENINIŲ VAIZDŲ KODAVIMĄ

Deividas Kančelkis

Programų sistemų inžinerijos katedra, informatikos fakultetas, KTU Studentų g. 50, Kaunas

Šiame straipsnyje pristatoma nauja lokaliai progresyvaus vaizdų kodavimo idėja (procedūra). Procedūra grindžiama specifinėmis Haaro bangelių savybėmis bei progresyviųjų vaizdų kodavimą realizuojančiu EZW algoritmu. Siūlomos lokaliai progresyvaus vaizdų kodavimo idėjos atsiradimą ir išvystymą sąlygojo du faktoriai — nuolat augančios kompiuterių kanalais (tinklais) perduodamos grafinės informacijos (vaizdų) apimtys bei ribotas kompiuterinių kanalų pralaidumas.

1. Įvadas

Pastaraisiais metais pastebimai išaugo kompiuterių panaudojimas įvairioms su informacijos apdorojimu susijusioms užduotims spręsti. Grafinės informacijos (skaitmeninių vaizdų) saugojimas, apdorojimas, perkėlimas iš vienos talpyklos į kitą tapo vienu iš bendriausių akcentų. Skaitmeninius vaizdus apimančios rinkmenos (failai) gali būti ganėtinai didelės ir gali užimti brangią (kompiuterio, kitos laikmenos) atminties vietą. Pavyzdžiui, skaitmeninį vaizdą 256 x 256 su pilka šviesos intensyvumo skale sudaro 65 536 elementai. Tipinis 640 x 480 dydžio spalvotas vaizdas turi beveik milijoną vaizdo elementų! Dėl tokios rinkmenų apimties, jų perdavimas internetu kartais tampa ilgu ir nuobodžiu procesu. Diskrečiųjų Haaro bangelių pagalba galima norimu santykiu suspausti vaizdą, kad jis užimtų mažiau talpos, o jo perdavimas elektroniniu būdu vyktų kur kas sparčiau.

2. Diskrečiosios Haaro transformacijos (HT) apskaičiavimas vaizdo fragmentams

Ortogonali Haaro funkcijų sistema (bangelės) paprastai pateikiama taip: $haar(0, 0, x) = 1, x \in [0, 1)$; $haar(r, m, x) = -\sqrt{2^r}$, kai $(m - 1) / 2^r \le x < (m - 1 / 2) / 2^r$; $haar(r, m, x) = -\sqrt{2^r}$, kai $(m - 1 / 2) / 2^r \le x < m / 2^r$; $haar(r, m, x) = 0$, kitu atveju; čia: $r = 0, 1, 2, ...$; $m = 1, 2, ..., 2^r$, [2]. Haaro (bangelių) skleidiniai turi daug įdomių savybių, kurios nėra būdingos kitokio tipo skleidiniams (Furjė, Volšo ir pan.). Viena jų yra lokalizavimo erdvėje savybė — jei domina skleidžiamos funkcijos (signalo) „elgesys“ intervale $[a;b] \subset [0,1)$, pakanka analizuoti tiktai nedidelį Haaro skleidinio koeficientų poaibį. Atkreiptinas dėmesys ir į tai, kad Haaro bangelės yra sukurtos iš vienos („motininės“) bangelės $haar(0, 1, x)$, $x \in [0, 1)$, naudojant diadinius mastelio keitinius ir sveikaskaitinius postūmius. Neabejotinai, lokalizavimas erdvėje ir pastaroji savybė yra Haaro (bangelių) skleidinių skiriamasis bruožas. Dėka šių savybių Haaro bangelės (Haaro skleidiniai, diskretusis Haaro spektras) sėkmingai panaudojamos skaitmeninių vaizdų (signalų) apdorojimo sferoje.

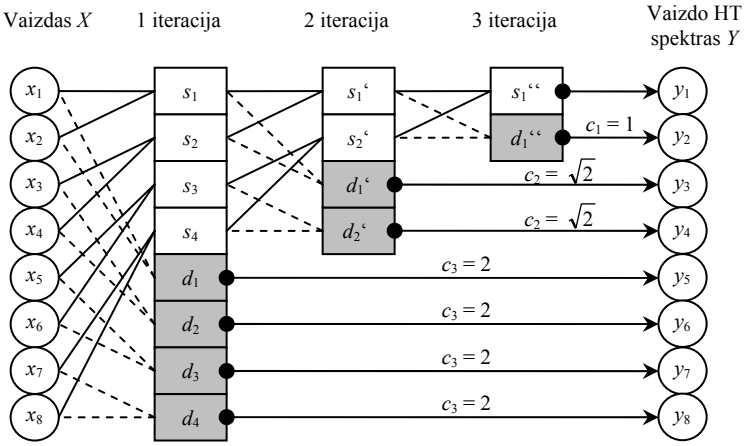
Žemiau trumpai pristatysime HT spektro apskaičiavimo skaitmeniniam vaizdui bei atskiriems to vaizdo fragmentams algoritmus.

Tarkime, turime vienmatį vektorių (skaitmeninį vaizdą) $X = (x_1, x_2, ..., x_N)^T$; $N = 2^n$, $n \in \boldsymbol{N}$. Haaro spektro vaizdui X skaičiavimas grindžiamas aritmetiniais sumų ir skirtumų vidurkiais. Jų gaunami tokie rezultatai: $s_k = (x_{2k-1} + x_{2k})/2$, $d_k = (x_{2k-1} - x_{2k})/2$; čia $k = 1,2,..., N/2$. Gaunami atitinkamai sumų ir skirtumų vidurkių vektoriai $s = (s_1, s_2, ..., s_{N/2})$ ir $d = (d_1, d_2, ..., d_{N/2})$. Formuojamas naujas vektorius $(s \mid d) = (s_1, s_2, ..., s_{N/2} \mid d_1, d_2, ..., d_{N/2})$, aprašantis Haaro transformacijos veikimo principą.

Vektorius (s | d) — tai tik pirmosios HT apskaičiavimo algoritmą realizuojančios iteracijos rezultatas. Norint gauti HT spektrą vienmačiam vaizdui (vektoriui) X, reiktų pakartoti šią procedūrą log_2N kartų, kaskart vietoj X imant naujai gaunamą vektorių s. Pastebėsime, kad po kiekvienos iteracijos vektoriaus d elementai, padauginti iš atitinkamų svorio koeficientų, interpretuojami kaip HT spektro poaibis, [1]. Geometrinė šio greito HT skaičiavimo algoritmo interpretacija (signalinis grafas; $N = 8$) pateikta 1 pav.

Dvimačiams skaitmeniniams vaizdams HT spektras apskaičiuojamas pakartotinai taikant vienmatę HT, t.y. pirmiausia kiekvienam pradinio vaizdo stulpeliui randamas HT spektras, po to gauto tarpinio masyvo eilutėms vėlgi taikoma vienmatė HT. Kitaip tarus, dvimačio vaizdo $N \times N$ HT spektrui gauti reikia $2N$ kartų panaudoti vienmatę Haaro transformaciją.

Kalbant apie lokaliai progresyvaus dvimačių skaitmeninių vaizdų kodavimo idėją, apie jos įgyvendinimą, ypatingą svarbą įgyja diskrečiojo HT spektro radimas atskiriems vaizdo fragmentams. Pasirodo, jog turint viso vaizdo HT spektrą, įmanoma operatyviai apskaičiuoti bet kurio vaizdo fragmento HT spektrą, ir tai reikalauja kur kas mažiau laikinių sąnaudų negu tiesioginis HT spektro apskaičiavimas atitinkamam vaizdo fragmentui.

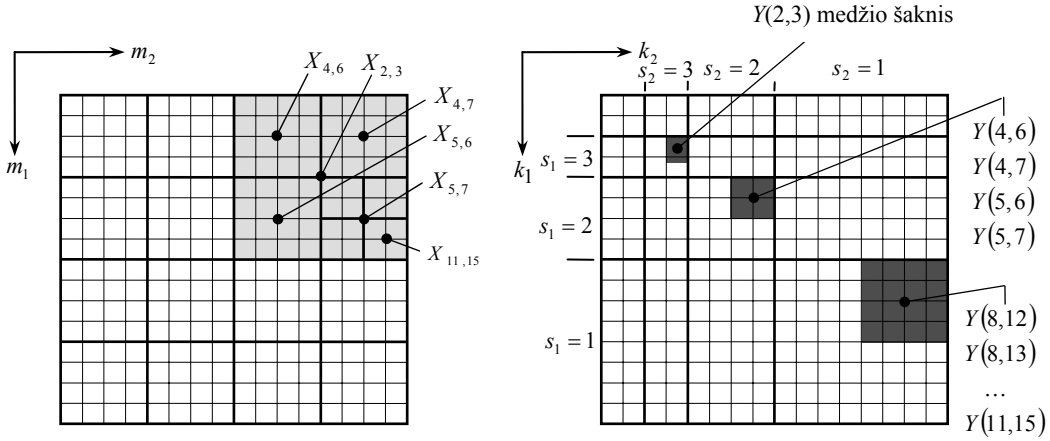


2.1 pav. HT skaičiavimo algoritmo signalinis grafas ($N = 8$)

Tarkime, kad $[X(m_1, m_2)]$ ir $[Y(k_1, k_2)]$ yra atitinamai dvimatis skaitmeninis vaizdas ir jo diskretusis Haaro (bangelių) spektras; $m_1, m_2, k_1, k_2 \in \{0,1,..., N-1\}$, $N = 2^n$, $n \in \boldsymbol{N}$. Kiekvienas spektrinis koeficientas $Y(k_1, k_2)$, $(k_1, k_2) \notin \{(0,0),(0,1),(1,0),(1,1)\}$, turi labai įdomias (praktiniu požiūriu) savybes, būtent:

1. Koeficiento $Y(k_1, k_2)$ $(k_i = 2^{n-s_i} + l_i, l_i \in \{0,1,...,2^{n-s_i} - 1\}, i = 1,2)$ skaitinę reikšmę nusako išskirtinai vaizdo $[X(m_1, m_2)]$ fragmentas (blokas) $X_{k_1,k_2} = [X(m_1, m_2) \mid (m_1, m_2) \in V_{k_1} \times V_{k_2}]$; čia $V_{k_i} = \{l_i \cdot 2^{s_i}, l_i \cdot 2^{s_i} + 1, ..., (l_i + 1) \cdot 2^{s_i} - 1\}$, $i = 1,2$ (2 pav., a).

2. Spektrinių koeficientų poaibis (medis) gali būti priskirtas koeficientui $Y(k_1, k_2)$, kai $s_1 > 1$ ir $s_2 > 1$. Medžio viršūnių (spektrinių koeficientų) aibė $\{Y(k_1^*, k_2^*) \mid (k_1^*, k_2^*) \in \bigcup_{t=1}^{\min\{s_1-1, s_2-1\}} (\Im_{k_1}(t) \times \Im_{k_2}(t))\}$; čia $\Im_{k_i}(t) = \{2^t \cdot k_i, 2^t \cdot k_i + 1, ..., 2^t \cdot (k_i + 1) - 1\}$, su visais $t = 1,2,..., s_i - 1$, $i = 1,2$. Pats koeficientas $Y(k_1, k_2)$ dažnai vadinamas medžio šaknimi (2 pav., b).



2.2 pav. Haaro spektro savybių grafinė interpretacija: (a) vaizdas [$X(m_1, m_2)$], $N = 16$, spektro koeficientas $Y(2, 3)$ susijęs su vaizdo fragmentu $X_{2,3}$; (b) koeficientui $Y(2, 3)$ priskirtas medis ($\Im_2 = \Im_2(1) \cup \Im_2(2) = \{4, 5, 8, 9, 10, 11\}$,

$$\Im_3 = \Im_3(1) \cup \Im_3(2) = \{6, 7, 12, 13, 14, 15\}$$ **)**

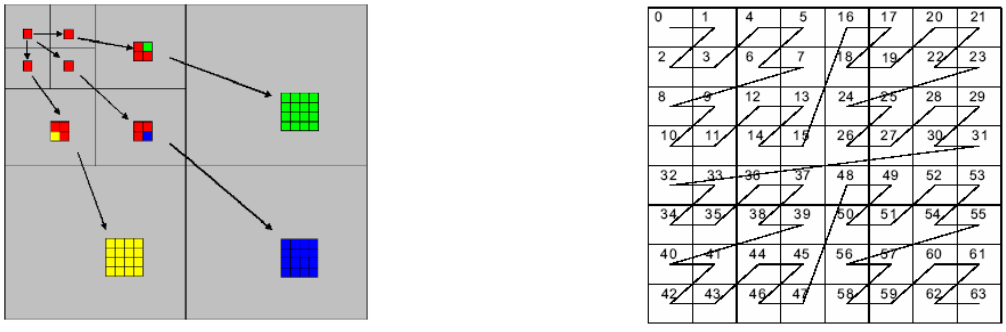
Remiantis šiomis savybėmis buvo sudarytas originalus greitas HT spektro apskaičiavimo vaizdo [$X(m_1, m_2)$] fragmentams X_{k_1,k_2} algoritmas, kuris, kaip parodė vėlesni teoriniai ir eksperimentiniai tyrimai, yra 10 ÷ 15 kartų spartesnis už tiesiogines HT spektro atitinkamiems vaizdo fragmentams apskaičiavimo procedūras.

Kadangi lokaliai progresyvaus vaizdų kodavimo idėja yra realizuota bangelių spektrinėje srityje, greitas perėjimas nuo viso vaizdo diskrečiojo Haaro spektro prie pasirinkto vaizdo bloko Haaro spektro, yra labai vertingas (vaidina lemiamą vaidmenį).

3. Lokaliai progresyvaus vaizdų kodavimo idėja

Lokaliai progresyvaus vaizdų kodavimo idėja (metodas) remiasi gerai žinomu progresyviojo vaizdų kodavimo algoritmu EZW (Embedded Zero-tree Wavelet) [3]. EZW algoritmas „dirba“ spektrinėje HT srityje ir grindžiamas medžių vaizdo HT spektre analizės rezultatais (3 pav., a). Lokaliai progresyvaus vaizdų kodavimo atveju, EZW algoritmas pirmiausia taikomas visam vaizdui. Gavus (po dekodavimo) „grubų“ vaizdo įvertį ir pasirinkus tolimesniam detalizavimui kurį nors vaizdo fragmentą (bloką), EZW algoritmas pakartotinai taikomas šiam blokui. Tai leidžia atskleisti naujas detales ne visame vaizde, o būtent pasirinktame bloke.

EZW algoritmas remiasi intuityvia nuostata — jeigu kurio nors HT spektrinio koeficiento (medžio šaknies) skaitinė reikšmė maža, tai labai tikėtina, jog kitų spektrinių koeficientų (to medžio viršūnių) skaitinės reikšmės irgi bus mažos. Kitaip tarus, atsiradus galimybei „ignoruoti“ medžio šaknį, tuo pačiu galime „ignoruoti“ ir visas medžio viršūnes (medis įvardijamas kaip „nulinis“).



(a) (b)

3.1 pav. Medžio struktūros ir jų skenavimo trajektorijos: (a) medžių struktūros (kiekvienas spektrinis koeficientas turi 4 spektrinius koeficientus aukštesniuose dažniuose); (b) Morton skenavimo kreivė

Trumpas EZW algoritmo aprašas pateikiamas žemiau.

1. Nustatomas pradinis koduojamo vaizdo [$X(m_1, m_2)$] spektro [$Y(k_1, k_2)$] skenavimo slenkstis: $t_0 = 2^{r_{\max}}$, kur $r_{\max}$ yra sveikoji skaičiaus $log_2(\max\{|Y(k_1, k_2)|\})$ dalis. Realizuojant algoritmą, slenkstis nuosekliai mažinamas, t.y. $t_k = t_0 / 2^k$, $k = 1, 2, \ldots, r_{\max}$.

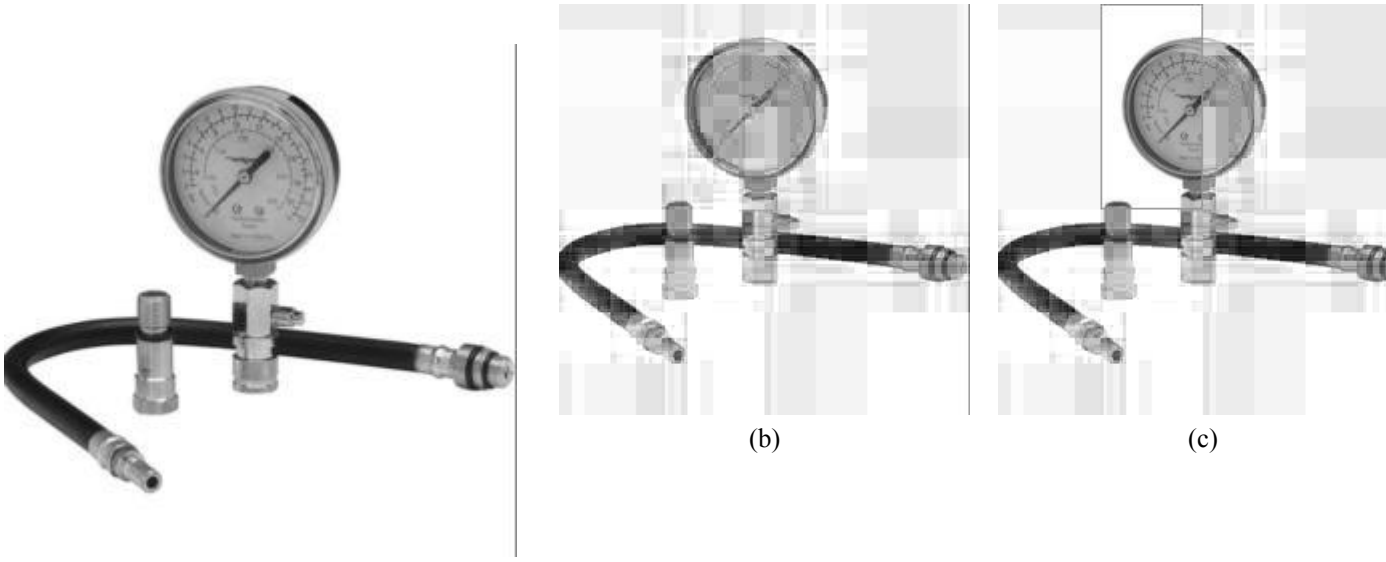
2. Su kiekviena slenksčio reikšme t_k, atliekamas vaizdo HT spektro skenavimas panaudojant Morton kreives (3 pav., b). Skenavimo metu spektro koeficientai $Y(k_1, k_2)$ yra užkoduojami: simboliu „*P*“ (jei koeficientas didesnis nei nurodytas slenkstis t_k), simboliu „*N*“ (jei koeficientas mažesnis už neigiamą slenkstį $-t_k$), simboliu „*T*“ (jei koeficientas yra nulinio medžio šaknis) arba simboliu „*Z*“ (jei koeficientas yra mažesnis nei slenkstis t_k, bet medis nėra nulinis). Koeficientai, užkoduoti simboliais „*P*“ ir „*N*“ yra šalinami iš spektro ir kaupiami FIFO masyve. Masyve sukauptos reikšmės koduojamos dvejetainiais kodais.

3. Antrasis punktas kartojamas iki $t_k \geq t_{\min}$.

Vaizdo atkūrimo (dekodavimo) etapo čia detaliau neaprašinėsime.

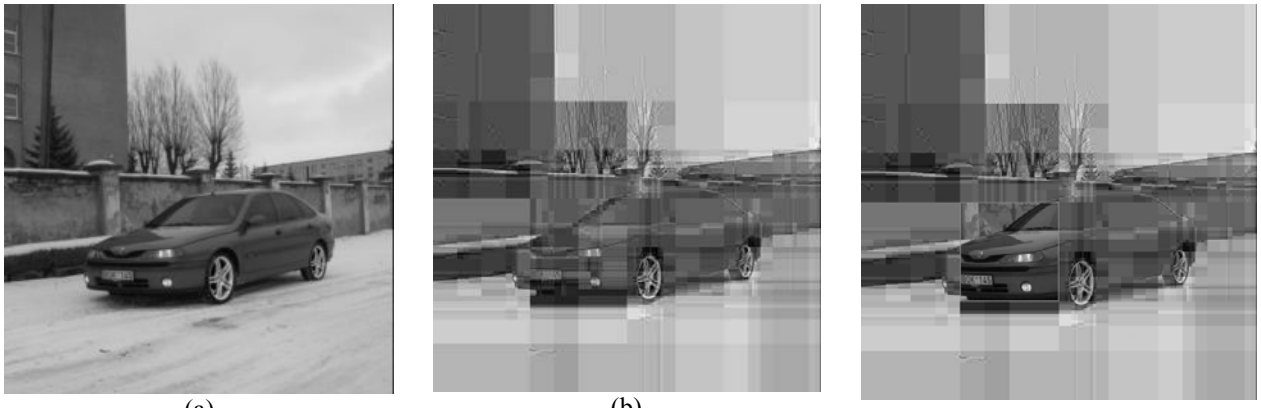
Kalbant apie pačią lokaliai progresyvaus vaizdų kodavimo idėją, būtina pabrėžti, jog ši idėja yra perspektyvi ir gyvybinga kai: (1) Vaizdo fragmentas yra tam tikra prasme susijęs su visu vaizdu kaip su kontekstu, t.y. be šio konteksto vaizdo fragmentas yra mažiau reikšmingas; (2) Atskiri vaizdo fragmentai pasižymi dideliu (išskirtiniu) informatyvumu, todėl jie turėtų būti atkurti (po dekodavimo) kiek įmanoma didesniu tikslumu. Bendras, santykinai aukštas, vaizdo suspaudimo efektas išgaunamas „stipriau“ spaudžiant visą apdorojamą vaizdą ir „švelniai“ spaudžiant pasirinktus atskirus vaizdo fragmentus. Pagrindiniai lokaliai progresyvaus vaizdų kodavimo etapai yra šie: 1) Taikant EZW algoritmą, koduojamas visas vaizdas (iki pasirinktos slenksčio reikšmės; „stiprus“ suspaudimas); 2) Pasirenkamas atkurto vaizdo fragmentas (po dekodavimo, t.y. gavus „grubų“ vaizdo įvertį). Apskaičiuojamas pasirinktojo fragmento HT spektras, taikant greitą skaičiavimo algoritmą. 3) Pakartotinai taikomas EZW algoritmas, tik šįkart vaizdo fragmentui (su mažesnėmis slenksčio reikšmėmis; „švelnus“ suspaudimas).

Tyrimo metu buvo realizuoti bazinis EZW algoritmas (naudota techninė įranga — Pentium IV 2,8 GHz, 512 MB RAM) ir greito HT apskaičiavimo vaizdo fragmentams procedūra bei atlikti preliminarūs eksperimentai (žr. 4, 5 pav.).



(a) (b) (c)

3.2 pav. Lokaliai progresyvaus skaitmeninių vaizdų kodavimo rezultatai: (a) originalus vaizdas *tester.bmp* 512 × 512; (b) atkurtas „grubus“ vaizdas, kai $t_{\min} = 8$; (c) progresyvaus kodavimo rezultatas (patikslintos detalės lokalizuotame vaizdo fragmente $X_{2,5}$)



(a) (b) (c)

3.3 pav. Lokaliai progresyvaus skaitmeninių vaizdų kodavimo rezultatai: (a) originalus vaizdas *laguna.bmp* 512 × 512; (b) atkurtas „grubus“ vaizdas, kai $t_{min} = 8$; (c) progresyvaus kodavimo rezultatas (patikslintos detalės lokalizuotame vaizdo fragmente $X_{6,5}$)

4. Išvados

Straipsnyje pateikiama visiškai nauja dvimačių skaitmeninių vaizdų lokaliai progresyvaus kodavimo idėja. Idėjos esmė – papildomi informacijos bitai leidžia atskleisti naujas detales ne į visą atkurtą vaizdą, o į (tiksliniai) pasirinktus atskirus atkurto vaizdo fragmentus.

Šią originalią vaizdų kodavimo idėją, be jokios abejonės, galima panaudoti ten, kur yra poreikis pasiųsti didelius grafinės informacijos kiekius riboto pralaidumo kanalais.

Kita vertus, siūlomą idėją galima adaptuoti dvimačiams spalvotiems bei vienmačiams vaizdams.

Literatūra

[1] **Chang, P; Piau P.** Modified Fast and Exact Algorithm for Fast Haar Transform. International Journal of Computer Science and Engineering, 2008, Vol. 2, No. 2, 55–58.

[2] **Shapiro J.M.** Embedded Image Coding Using Zero-trees of Wavelet Coefficients. IEEE Transactions on Signal Processing, 1993, Vol. 41, No. 12, 3445–3462.

[3] **Valantinas J.** On the application of Haar wavelet to locally progressive encoding of grey-level images. ISSN 1392 – 124X Information technology and control, 2007, Vol. 36, No. 2, 177–186.

On the locally progressive encoding of digital images

In this paper, a novel locally progressive image encoding idea (procedure) is presented. The procedure explores both specific properties of Haar wavelets and the EZW algorithm used for progressive image encoding. It is true to say that the rise and development of idea determined two factors — increasing amounts of graphical information (images) transmitted via computer channels (networks) and limited capacity the latter.