

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMATIKOS STUDIJŲ PROGRAMA

ANDRIUS AMELYNAS

**EFEKTYVAUS ENTROPINIO KODAVIMO ALGORITMO
SUDARYMAS IR TAIKYMAS JPEG STANDARTE**

Magistro darbas

Darbo vadovas
doc. dr. A. Ostreika

KAUNAS, 2013

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMATIKOS STUDIJŲ PROGRAMA

ANDRIUS AMELYNAS

EFEKTYVAUS ENTROPINIO KODAVIMO ALGORITMO
SUDARYMAS IR TAIKYMAS JPEG STANDARTE

Magistro darbas

Darbo vadovas:
doc. dr. A. Ostreika
2013-05-

Recenzentas:
prof. dr. B. Paradauskas
2013-05-

Atliko:
IFM-1/1 gr. studentas
Andrius Amelynas
2013-05-23

KAUNAS, 2013

AUTORIŲ GARANTINIS RAŠTAS

DĖL PATEIKIAMO KŪRINIO

2013 - 05 - 20 d.
Kaunas

Autorius,

Andrius Amelynas,

(vardas, pavardė)

patvirtina, kad Kauno technologijos universitetui pateiktas baigiamasis bakalauro (magistro) darbas (toliau vadinama – Kūrinys) Efektyvaus entropinio kodavimo algoritmo sudarymas ir taikymas
(kūrinio pavadinimas)

JPEG standarte,

pagal Lietuvos Respublikos autorių ir gretutinių teisių įstatymą yra originalus ir užtikrina, kad

- 1) jį sukūrė ir parašė Kūrinyje įvardyti autoriai;
- 2) Kūrinys nėra ir nebus įteiktas kitoms institucijoms (universitetams) (tiek lietuvių, tiek užsienio kalba);
- 3) Kūrinyje nėra teiginių, neatitinkančių tikrovės, ar medžiagos, kuri galėtų pažeisti kito fizinio ar juridinio asmens intelektinės nuosavybės teises, leidėjų bei finansuotojų reikalavimus ir sąlygas;
- 4) visi Kūrinyje naudojami šaltiniai yra cituojami (su nuoroda į pirminį šaltinį ir autorių);
- 5) neprieštarauja dėl Kūrinio platinimo visomis oficialiomis sklaidos priemonėmis.
- 6) atlygins Kauno technologijos universitetui ir tretiesiems asmenims žalą ir nuostolius, atsiradusius dėl pažeidimų, susijusių su aukščiau išvardintų Autorių garantijų nesilaikymu;
- 7) Autoriai už šiame rašte pateiktos informacijos teisingumą atsako Lietuvos Respublikos įstatymų nustatyta tvarka.

Autoriai

Andrius Amelynas

(vardas, pavardė)

(parašas)

(vardas, pavardė)

(parašas)

(vardas, pavardė)

(parašas)

(vardas, pavardė)

(parašas)

SANTRAUKA

Kompiuteriams pradėjus saugoti duomenis, atsirado poreikis ir mažinti tų saugojamų duomenų apimtį. Nesuglaudinti duomenys užima daug vietos- iškyla duomenų saugojimui skirtos atminties talpos problema, be to tokių duomenų perdavimas trunka ilgiau. Šie faktai nesuderinami su dabartinėmis technologijomis, nes šiuo metu visose srityse naudojamas vienokio ar kitokio tipo duomenų glaudinimas. Yra du pagrindiniai glaudinimo tipai- su informacijos praradimu ir be jo. Šiame darbe, vaizdams glaudinti, naudojamas algoritmas su informacijos praradimu- JPEG. Šio standarto kontekste, analizuojami duomenų sekoms koduoti skirti entropiniai algoritmai be informacijos praradimo.

Šio darbo tikslas yra sukurti ir aprašyti duomenų sekos glaudinimo algoritmą, palyginant jį su simbolių sekos ilgio kodavimo RLE algoritmais, taip išskiriant privalumus ir trūkumus, bei pateikiant rekomendacijas jo panaudojimui. Tyrimas atliekamas panaudojant JPEG standartą, analizuojant vaizdų ar jų fragmentų duomenų bazę.

Išanalizavus visus gautus rezultatus, gauta išvada, jog sukurtasis algoritmas yra vidutiniškai 16,56% efektyvesnis, nei abi RLE algoritmo versijos. A&A algoritmo suglaudinti duomenys, vidutiniškai užimdavo 14,58% nesuglaudintų duomenų apimties. Dėl išvardintų privalumų, A&A algoritmą galima rekomenduoti naudoti tiek vietoje JPEG naudojamos RLE modifikacijos, tiek ir kitose srityse, pasikartojančių simbolių sekų ilgiui koduoti.

SUMMARY

Since computers began to store information, the need to make smaller capacity of stored data appears. Raw data occupies quite a lot of space in computer memory, and of course the transmissions of such data are longer. These facts are incompatible with the current technology, which is currently used in all areas of one or another type of data compression. There are two main types of compression -with information loss or without it. In this paper, the algorithm used to compress images with data loss - JPEG. This standard in the context of analyzing the data sequences coding for the entropic algorithms without loss of information.

The main purpose in this paper is to develop and describe the sequence data compression algorithm, compare it with the character sequence length RLE encoding algorithms, and to find put the advantages and disadvantages of this type compression. And to write recommendations to use it. The survey is conducted using the JPEG standard, the analysis of images or portions of the database.

The analysis of all the results obtained by the finding that the created algorithm is on average 16.56% more effective than the both versions of RLE algorithm. A & A data compression algorithm, on average, 14.58% occupied uncompressed data volume. The following advantages of A & A algorithm to recommend the use of both on-site use JPEG RLE modification, and in other areas of recurring characters in length coding sequences.

TURINYS

LENTELIŲ SĄRAŠAS.....	8
PAVEIKSLŲ SĄRAŠAS.....	9
TERMINŲ IR SANTRUMPŲ ŽODYNAS	11
ĮVADAS	13
1. DVIMAČIO VAIZDO KODAVIMO METODŲ ANALIZĖ.....	15
1.1. Skaitmeninio vaizdo ir jo glaudinimo metodų analizės tikslas.....	15
1.2. Skaitmeninio vaizdo samprata	15
1.2.1. Su skaitmeniniu vaizdu susijusios sąvokos	16
1.2.2. Skaitmeninių vaizdų tipai	16
1.3. Informacijos glaudinimo algoritmų poreikis	16
1.4. Glaudinimo principai.....	17
1.4.1. Erdvinis pertekliškumas	17
1.4.2. Kodavimo pertekliškumas	18
1.4.3. Spalvinis pertekliškumas.....	19
1.4.4. Dažninis pertekliškumas	19
1.4.5. Kodavimas.....	20
1.5. Standartinė vaizdo kodavimo schema	20
1.6. Glaudinimo algoritmų charakteristikos	21
1.6.1. Glaudinimo lygis.....	21
1.6.2. Glaudinimo greitis	21
1.6.3. Glaudinto vaizdo kokybė.....	21
1.7. Plačiausiai paplitę iškraipantys vaizdų glaudinimo metodai	22
1.7.1. BTC algoritmas.....	22
1.7.2. JPEG kodavimo algoritmas	23
1.8. Duomenų entropija.....	31
1.8.1. Šenono entropija ir teorema.....	31
1.8.2. Algoritmų sudėtingumas	32
1.8.3. Algoritmų tinkamumas.....	33
1.9. Hafmano kodavimo algoritmas	33
1.10. LZW kodavimo algoritmas	35
1.11. Run-Length Encoding algoritmas	38
1.11.1. Sekos ilgio kodavimo variantai.....	39
1.11.2. Sekos ilgio kodavimo specifika	40
1.11.3. Kryžminis kodavimas.....	41
1.11.4. Bitų, baitų ir pikselio lygio RLE schemas	41
1.12. Sudaromo A&A algoritmo naudotojų analizė	44
1.13. Esamų kodavimo pertekliškumo mažinimo algoritmų analizė	45
1.14. A&A algoritmas	45

1.15. Skaitmeninio vaizdo ir jo glaudinimo metodų analizės išvados	45
2. EKSPERIMENTINĖS PROGRAMINĖS ĮRANGOS REIKALAVIMŲ SPECIFIKACIJA IR A&A ALGORITMO APRAŠAS.....	47
2.1. Eksperimentinė programinė įranga	47
2.2. Funkciniai ir nefunkciniai reikalavimai.....	47
2.3. Algoritmo sudarymas	47
3. A&A ALGORITMO REALIZACIJA IR TYRIMAS	54
3.1. Realizacijos modelis.....	54
3.2. Pradiniai duomenys	54
3.3. Algoritmų veikimo patikra.....	54
3.4. Algoritmų efektyvumo patikra.....	55
3.5. Rezultatai	57
3.5.1. Pirmos dalies bandymo rezultatai	57
3.5.2. Antros dalies bandymo rezultatai	60
3.5.3. Trečios dalies bandymo rezultatai	61
3.5.4. Apibendrinti bandymų rezultatai	62
4. IŠVADOS	68
5. LITERATŪRA	69
6. PRIEDAI	70
6.1. priedas. Programinės įrangos kodas	70
6.2. priedas. Pirmos dalies bandymų rezultatai	75
6.3. priedas. Antros dalies bandymų rezultatai.....	77
6.4. priedas. Trečios dalies bandymų rezultatai.....	79

LENTELIŲ SĄRAŠAS

1.1 lentelė Skirtingos duomenų apimties siuntimo laikas el. paštu	17
1.2 lentelė Dvejetainio kodavimo pavyzdys.....	18
1.3 lentelė Esamų entropinių kodavimo algoritmų palyginimas	45
3.1 lentelė Analizuojamų paveikslėlių kiekis ir tipai.....	54
3.2 lentelė Bylų, suglaudintų JPEG standartu (glaudinimo lygis 50%), naudojant abu RLE algoritmus ir A&A algoritmą palyginimas	57
3.3 lentelė Algoritmų tarpusavio palyginimas (kvantavimo matrica 50%)	60
3.4 lentelė Algoritmų tarpusavio palyginimas (kvantavimo matrica 70%)	61
3.5 lentelė Algoritmų tarpusavio palyginimas (kvantavimo matrica 30%)	62
3.6 lentelė Vidutinio algoritmų efektyvumo palyginimas su A&A algoritmu efektyvumu	63
3.7 lentelė Blogiausių ir geriausių algoritmų glaudinimo rezultatų suvestinė	63
3.8 lentelė Glaudinimo efektyvumas pagal paveikslėlių tipus, lyginant su vidutiniu duomenų suglaudinimo efektyvumu.	65

PAVEIKSLŲ SĄRAŠAS

1.1 pav. Skaitmeninė dvimatė vaizdo matrica.....	15
1.2 pav. Vaizdų gretimų taškų spalvų skirtumų histograma. X ašis- įvykio tikimybė, Y ašis- skirtumo reikšmė	18
1.3 pav. Gama korekcija. X ašis- Video signalo įtampa (mV). Y ašis- šviesos ryškumas ($cd * m^2$) .	19
1.4 pav. Gama korekcijos įtaka vaizdai (www.wikipedia.org)	19
1.5 pav. Standartinė vaizdo kodavimo schema.....	20
1.6 pav. Nesuglaudintas dvispalvis paveikslėlis (www.opticalengineering.spiedigitalibrary.org)	23
1.7 pav. Suglaudintas BTC algoritmu dvispalvis paveikslėlis (www.opticalengineering.spiedigitalibrary.org).....	23
1.8 pav. JPEG kodavimo algoritmas.....	24
1.9 pav. Chromatiškumo raiškos mažinimo schemas ir koeficientai	25
1.10 pav. Vaizdo duomenų blokas (www.wikipedia.org).....	26
1.11 pav. Vaizdo duomenų bloko reikšmės po perslinkimo (www.wikipedia.org)	26
1.13 pav. Tipinė (50 %) skaisčio (komponentės Y') kvantavimo matrica (www.ams.org) .	27
1.14 pav. Tipinė (50 %) chromatiškumo (komponenčių CB ir CR) kvantavimo matrica (www.ams.org) .	28
1.15 pav. Sekos iš matricos sudarymas zigzago principu.	28
1.16 pav. 100 proc. kokybės suglaudintas paveikslėlis (www.wikipedia.org)	29
1.17 pav. 60 proc. kokybės JPEG suglaudintas paveikslėlis.....	29
1.18 pav. 10 proc. kokybės JPEG suglaudintas paveikslėlis.....	30
1.19 pav. JPEG failo fragmentas	30
1.20 pav. Hafmano medis (www.wikipedia.org).....	34
1.21 pav. Hafmano medžio duomenų lentelė	34
1.22 pav. Pradinė (įėjimo sekos) kodų vertimo lentelė.....	36
1.23 pav. Pavyzdinė LZW algoritmo kodavimo schema	37
1.24 pav. Pavyzdinė LZW algoritmo dekodavimo schema	37
1.25 pav. Nekoduota „A“ simbolių seka.....	38
1.26 pav. RLE koduota „A“ simbolių seka	38
1.27 pav. Nekoduota „A, B, C ir D“ simbolių seka.....	38
1.28 pav. RLE koduota „A, B, C ir D“ simbolių seka	39
1.29 pav. Seka prieš RLE kodavimą.....	39
1.30 pav. Seka po neefektyvaus RLE kodavimo	39
1.31 pav. RLE sekų sudarymo variantai	40
1.32 pav. Bazinė RLE algoritmo sekų diagrama	42
1.33 pav. Bitų, baitų ir pikselio lygio RLE schemas.	44
2.2 pav. A&A algoritmu užkoduota 11-os skaičių seka.....	48
2.3 pav. A&A algoritmo sekų diagrama	49

2.4 pav. JPEG 8x8 blokas po kvantavimo(www.ams.org).....	52
3.1 pav. 16x16 pikselių dydžio paveikslėlis (proporcijos neatitinka originalo).....	55
3.2 pav. Programinės įrangos išvesties kodas, RLE koduota seka	56
3.3 pav. Programinės įrangos išvesties kodas, modifikuotu RLE koduota seka	56
3.4 pav. Programinės įrangos išvesties kodas, A&A koduota seka.....	56
3.5 pav. Algoritmo efektyvumo palyginimas pagal paveikslėlių tipus.....	64
3.6 pav. Glaudinimo efektyvumo pagal paveikslėlių tipus palyginimas su vidutiniu duomenų suglaudinimo efektyvumu	66
3.7 pav. A&A algoritmo kodavimo efektyvumas, esant skirtingiems glaudinimo laipsniams	67

TERMINŲ IR SANTRUMPŲ ŽODYNAS

Be informacijos praradimo	(<i>angl. lossless</i>) tai toks duomenų suglaudimas, kai užkoduotas ir atkoduotas vaizdas yra identiški.
Su informacijos praradimu	(<i>angl. lossy</i>) tai toks duomenų suglaudimas, kai užkoduotas ir atkoduotas vaizdas nėra identiški, pradinis vaizdas iškraipomas.
Vizualiai identiški	(<i>angl. visually lossless</i>) tai kodavimo rezultate gautas rekonstruotas vaizdas, kuris vizualiai identiškas pradiniam, nors skaitiškai gali skirtis.
Vaizdo išplaukimas	(<i>angl. blur</i>) tai miglotas, išplaukęs vaizdas aplink fokusuojamą objektą.
JPEG	(<i>ang. Joint Photographic Experts Group</i>) standarto pavadinimas tai angliškų žodžių santrumpa.
RLE	(<i>angl. Run-length encoding</i>) bitų sekos kodavimas.
A&A	Audriaus Amelyno algoritmas. Kitaip A algoritmas.
RGB	(<i>angl. Red, Green Blue</i>), spalvų užrašymo metodas (erdvė).
$Y C_B C_R$	(<i>angl. $Y C_B C_R$ Y – luminance, C - chroma (Blue, Red)</i>). Spalvų užrašymo metodas (erdvė).
Entropija	(<i>angl. Entropy</i>). Entropijos sąvoka duomenų glaudinime (kodavime) naudojama žymėti glaudinimo algoritmo įėjimo duomenų atsitiktinį išsidėstymą (netvarkingumą, chaotiškumą). Kuo didesnis entropijos laipsnis, tuo mažesnis glaudinimo efektyvumas.
Buferis	Atminties dydis dinaminėje atmintinėje (<i>angl. RAM- Random Access Memory</i> , liet. atsitiktinės kreipties atmintis), naudojamas laikinam įėjimo ar išėjimo duomenų saugojimui.
Aplikacija	(<i>angl. application</i>). Programinė įranga, taikomoji programa.
Paraidinė (seka)	(<i>angl. literal (run)</i>). Taip vadinama nekoduota simbolių seka koduotoje sekoje. Šiuo atveju nekoduoti simboliai eina nuosekliai ir tokia pat tvarka, kaip ir įėjimo sekoje.
Piksėlis	(<i>angl. pixel</i>). Dvimatis skaitmeninio vaizdo elementas. Jį gali sudaryti keli taškai.
Vokselis	(<i>angl. voxel</i>). Erdvinis vaizdo elementas, lygus pikseliui, padaugintam iš pjūvio storio.
Taškas	(<i>angl. dot</i>). Mažiausias skaitmeninio vaizdo elementas.
BMP	(<i>angl. Bit Map</i>). Grafikos vaizdų failo formatas, failo plėtinys. Angliškų žodžių santrumpa. Lietuviškai- bitų žemėlapis.
Dvejetainis	(<i>angl. binary</i>) [lot. binarius- dvigubas], dvinaris, dvilypis, susidedantis iš 2 dalių; paremtas tų dalių (narių) gretinimu, lyginimu, priešprieša. Informatikoje naudojamas apibrėžti dvejetainiams skaičiams, susidedantiems iš 1 ir 0.
Gama korekcija	(<i>angl. gamma correction</i>). Tai netiesinė (kėlimo laipsniu funkcija), skirta (de)koduoti šviesumą (šviesos intensyvumą) arba spalvos ryškumą.
Chroma (C)	Atspalvis, nustatomas mėlynos ir raudonos spalvos skirtumo pagalba.
Luminance (Y)	Šviesumas, šviesos intensyvumas.
Luma (Y')	Šviesumas su gama korekcija (' žymi gama korekciją)
BTC	(<i>angl. Block Truncation Coding</i>). Angliškų žodžių santrumpa. Vaizdų glaudinimo

	algoritmas, su informacijos praradimu.
DCT	(<i>angl. Discrete Cosine Transform</i>) Diskrečioji kosinuso transformacija.
Kodekas	(<i>angl. Codec</i>) Sudurtinis žodis iš anglišku žodžių <i>Code</i> (<i>koduoti</i>) ir <i>Decode</i> (<i>dekoduoti</i>).
LZW	(<i>angl. Lempel–Ziv–Welch</i>) žodžių santrumpa. Entropinis žodyninis glaudinimo metodas.
Klasteris	(<i>angl. cluster</i>) vienetas, sudarytas iš kelių vieno tipo elementų, kuris veikia kaip vieninga sistema, turinti tam tikrų savybių.

IVADAS

Kompiuterio funkcijoms pasipildžius viena papildoma – saugoti duomenis išorinėse laikmenose, o ne tik juos išvesti į ekraną ar spausdintuvą, atsirado poreikis ir mažinti tų saugojamų duomenų apimtį. Taškiniai vaizdai ne išimtis, nes būtent jiems saugoti sunaudojama didesnė dalis vietos pastovioje atminties talpykloje. Ši problema ir dabar ne tik neišnyko, bet tapo dar labiau aktuali, nes su dabartinėmis grafinių vaizdų bei kitomis programomis sukurtus taškinius vaizdus, modifikuotas nuotraukas ar kitokius paveikslėlius norime išsaugoti kuo geresnės kokybės ir žinoma kuo mažesnės apimties faile. Ši problema nėra nauja ir sprendimai jai modeliuojami jau seniai.

Galima išskirti dvi pagrindines duomenų glaudinimo algoritmų klases- glaudinimo algoritmai be informacijos praradimo (angl. lossless) ir algoritmai su duomenų praradimu (angl. lossy). Naudojant algoritmus be informacijos praradimo, rekonstruotas vaizdas yra skaitiškai identiškas originaliam vaizdui. Tačiau, naudojant šio tipo algoritmus pasiekiamas tik nežymus duomenų glaudinimo lygis. Kai naudojami algoritmai su duomenų praradimu, atkurtas vaizdas netiksliai atkartoja originalų vaizdą. Informacijos praradimas vyksta todėl, kad glaudinimo metu yra visiškai atmetama perteklinė (mažiausiai svarbi) informacija. Šis būdas leidžia daug didesnius duomenų glaudinimo lygius. Net ir esant skaitiniams duomenų praradimams, dažnai rekonstruotas ir pradinis vaizdas yra vizualiai identiški (angl. visually lossless).

Yra daug standartų, aprašančių įvairių duomenų tipų glaudinimo algoritmus, eiliškumą ir tvarką. Mūsų atveju aktualūs yra standartai, nusakantys statinio vaizdo kodavimo algoritmus ir tvarką, nurodantys duomenų kodavimo metodus po vaizdo duomenų glaudinimo. Žinomiausias yra seniai naudojamas ir plačiausiai paplitęs formatas- JPEG. Jį 1992 metais publikavo jungtinė fotografų ekspertų grupė (angl. *Joint Photographic Experts Group*).

Darbo problematika ir aktualumas

Norint sumažinti vaizdo bylų apimtį, tam naudojant JPEG algoritmą, bet išlaikyti aukštą vaizdo kokybę, reikia rinktis nedidelį glaudinimo lygį. Tokiu atveju mažiau išnaudojami JPEG teikiami privalumai, tačiau bylų apimties sumažinimui, dar galima pasitelkti ir entropinius algoritmus. Pastarieji nekoduotos informacijos apimtį gali suglaudinti 80 ir daugiau procentų, visiškai neprarandant glaudinamų duomenų. Entropinių glaudinimo algoritmų efektyvumas žymiai skiriasi ir priklauso nuo koduojamų duomenų tipo. Norint pasiekti aukščiausią efektyvumą, toks kodavimo algoritmas turi būti ir labai efektyvus ir tuo pačiu universalus savo panaudojimo srityje.

Darbo tikslas ir uždaviniai

Šiame darbe, kaip duomenų šaltiniu bus remiamasi JPEG standartu, jis bus tobulinamas, mėginant padidinti standarto efektyvumą, duomenų suglaudimui panaudojant realizuotą simbolių sekos ilgio kodavimo algoritmą.

Šio darbo tikslas yra sukurti efektyvų (efektyvesnį už šiuo metu naudojamą RLE) duomenų sekos kodavimo algoritmą, pritaikant jį po JPEG standarte aprašyto dvimačių vaizdų glaudinimo gautoms duomenų sekoms užkoduoti.

Tikslas išskaidytas į tokius uždavinius:

1. Naudojant vaizdo ir jo fragmentų duomenų bazę, eksperimentiškai patikrinti sudaryto kodavimo algoritmo naudingumą, palyginant jį su RLE ir jo atmainomis, naudojamomis JPEG standarte.
2. Išanalizuoti gautus rezultatus bei ištirti sudarytojo algoritmo privalumus bei trūkumus.
3. Išvardinti galimus patobulinimus.
4. Integruoti į JPEG standartą, kaip papildymą didesnės spalvinės raiškos vaizdams koduoti, esant mažesniai vaizdo glaudinimo laipsniui.

Algoritmas turėtų pagerinti didelės apimties statinio vaizdo duomenų suglaudimą vidutiniškai bent 5 proc., palyginus su RLE algoritmu ir jo atmainomis. Duomenų glaudinimo laikas neturėtų labai skirtis nuo RLE ir kitų populiariausių duomenų glaudinimo algoritmų.

Numatomi darbo rezultatai ir jų svarba

Kuriamas A&A algoritmas turėtų būti efektyvus bet kokių tipų paveikslukų duomenų sekoms koduoti, nepriklausomai nuo vaizdo suglaudimo lygio. Šis algoritmas neturėtų padidinti

koduojamų duomenų apimties ir būti universalus panaudojimo atžvilgiu. Didžiausias pranašumas, lyginant su šiuo metu JPEG naudojamais RLE algoritmais, turėtų būti pasiekiamas esant mažam suglaudavimo lygiui, kuomet reikia išlaikyti gerą vaizdo kokybę ir maksimaliai sumažinti vaizdo bylos dydį.

Darbo struktūra

Šiame darbe pristatomas duomenų sekos glaudinimo algoritmas, naudojantis vaizdų ar jų fragmentų duomenų bazę, kuriai sudaryti naudojami spalviškai įvairių paveikslėlių klasterizavimo (klasifikavimo, grupavimo) metodai. Aprašytasis algoritmas tinka tiek pilkų (angl. greyscale) tiek spalvotų (ang. color) vaizdų (paveikslėlių) duomenų sekoms koduoti, nors daroma prielaida, jog efektyvesnis naudoti didelės spalvinės raiškos paveikslėlių duomenų sekoms. Esant didelės spalvinės raiškos paveikslėlių duomenų bazei bei naudojant mažesnę (mažiau kaip 50 %) glaudinimo laipsnį, bus pasiekiami geresni rezultatai, nei naudojant RLE. Vaizdo baitų sekos glaudinimo koeficientas yra priklausomas nuo vaizdo pobūdžio ir nuo duomenų sekai suglaudinti naudojamų parametrų.

Darbe yra apžvelgiami keli populiariausi vaizdo duomenų glaudinimo su informacijos praradimu algoritmai, skaičių sekos (be informacijos praradimo) kodavimo algoritmai, taip pat atlikta pastarųjų algoritmų privalumų bei trūkumų analizė. Taip pat išanalizuoti algoritmai, naudojami bitų sekoms koduoti pačiame JPEG standarte ir jo atmainose, bei kituose grafikos vaizdų failų formatuose. Pastaruosiuose gali būti glaudinami vaizdo duomenys, naudojami tik sekų kodavimo (ir) arba aritmetiniai kodavimai.

Išanalizavus visus gautus rezultatus, darbe pateikiami naujojo algoritmo privalumai bei trūkumai, vystymo perspektyvos ir galimi tolesni moksliniai tyrimai bei eksperimentai.

1. DVIMAČIO VAIZDO KODAVIMO METODŲ ANALIZĖ

Darbo pagrindas- simbolių sekos ilgio kodavimo algoritmo sukūrimas ir taikymas skaitmeninių dvimačių statinių vaizdų kodavime, JPEG standarto kontekste. Skaitmeninis vaizdas yra „realaus“ objekto atvaizdavimas dvimačiu (trimačiu) skaičių masyvu, sudarytu iš mažiausių vaizdo elementų- taškų, pikselių ar vokselių. JPEG nėra grafikos vaizdų formatas, tai tam tikras glaudinimo algoritmas, apibrėžiantis, kaip 2D statinis vaizdas transformuojamas į baitų srautą ir atgal į vaizdą, tačiau nenurodo failo struktūros. Toliau šiame skyriuje apžvelgiama kitų autorių pasiūlyti metodai ir kita susijusi literatūra.

1.1. Skaitmeninio vaizdo ir jo glaudinimo metodų analizės tikslas

Entropinio kodavimo algoritmams nėra būtina koduoti būtent vaizdo elementų duomenų sekas, tačiau pačiam kodavimui šaltinis būtinas. Informacijos teorijoje procesas, sukeliantis atsitiktinius įvykius, vadinamas šaltiniu. Be šaltinio pats kodavimas netenka prasmės. Darbe pasirinktu atveju, tokių sekų šaltinis yra JPEG suglaudintų vaizdų kvantuotų matricių elementai-duomenų sekos. Tam, kad suprastume kas yra skaitmeninis vaizdas, kokie dydžiai jį apibudina ir kaip veikia tokių vaizdų glaudinimo algoritmai, atliksime analizę.

1.2. Skaitmeninio vaizdo samprata

Pats skaitmeninis vaizdas apibūdinamas, kaip dvimatė reikšmių seka (1):

$$x[n_1, n_2]; \tag{1}$$

$$0 \leq n_1 < N_1, 0 \leq n_2 < N_2;$$

čia $x[n_1, n_2]$ – konkretaus vaizdo pikselio (taško) reikšmė, N_1 – vaizdo aukštis pikseliais; N_2 – vaizdo plotis pikseliais.

Gaunama skaitmeninė matrica: skaičių eilutės ir stulpeliai. Skaitmeninių dvimačių vaizdo matricių (1.1 pav.), pateiktų A, B ir C skiltyse kiekvienas matricos elementas atitinka vaizdo elementą (piksely).

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0, M - 1) \\ f(1,0) & \dots & \dots & f(1, M - 1) \\ \vdots & \vdots & \vdots & \vdots \\ f(N - 1, 0) & f(N - 1, 1) & \dots & f(N - 1, M - 1) \end{bmatrix} \tag{A}$$

0	0.75	0	0.75	0
1	0.25	1	0.25	1
0.5	0.75	0.5	0.75	0.5

(B)


1.1 pav. Skaitmeninė dvimatė vaizdo matrica

1.2.1. Su skaitmeniniu vaizdu susijusios sąvokos

- Skaitmeninės grafikos tipai skirstomi į:
 - taškinis (vaizdą sudaro dvimatė fiksuoto dydžio spalvotų taškų/ pikselių (*angl. pixel*) matrica, vienas taškas gali turėti vieną iš daugelio reikšmių, reprezentuojančių jo spalvą ir intensyvumą)
 - ir vektorinius vaizdus (sudaryti iš primityvių geometrinių objektų, kontūrai gali būti užpildomi spalva, spalvų gradientu (perėjimu) ar pasirinktu raštu) [3].
- Raiška (*angl. resolution*) gali būti apibrėžiama tokiais parametrais [3]:
 - Matricos matmenys: $H \times V$ (eilutės \times stulpeliai);
 - Kraštinių santykis: $H:V$;
 - Pikselių kiekis: MP;
 - Taškų tankis: DPI;
 - Pikselių tankis: PPI.
- Spalvos modeliai [3]:
 - RGB modelis: principas artimiausias žmogaus gebėjimui matyti spalvas, erdvę sudaro kubas.
 - HSL ir HSV modeliai: alternatyvūs modeliai, atskiriantys spalvą nuo intensyvumo, erdvę sudaro cilindras (kūgis).
 - CMYK modelis: modelis, atvirkštinis RGB modeliui, naudojamas spaudoje.
 - $Y C_B C_R$, $Y' C_B C_R$, YIQ, $Y D_B D_R$ modeliai: modeliai atskiriantys spalvą nuo intensyvumo, naudojami vaizdų kodavime ir TV transliacijose.
- Standartizuojant skaitmeninius vaizdus, galima išskirti keletą dalykų [3]:
 - Spalvų kiekis;
 - Kodavimo ir dekodavimo algoritmas;
 - Saugojimo formatas.
- Vaizdų kodavimo standartus reglamentuojančios organizacijos [3]:
 - JPEG (*angl. Joint Photographic Experts Group*);
 - JBIG (*angl. Joint Bi-level Image Experts Group*);
 - IESG (*angl. Internet Engineering Steering Group*).

1.2.2. Skaitmeninių vaizdų tipai

- Taškiniai intensyvumo vaizdai [1]:
 - pikselio reikšmė atitinką signalo intensyvumą.
- Grafiniai taškiniai paveikslėliai [1]:
 - Binariniai (dvejetainiai)- loginių reikšmių (0,1) masyvas;
 - RGB- pikselio reikšmė tai spalvos kodas;
 - Indeksuoti- pikselio reikšmė tai nuoroda į spalvos kodą.
- Grafiniai vektoriniai paveikslėliai [1]
 - Sudaromi ne iš pikselių, bet naudojant vaizdo formavimo komandas (geometriniai elementai ir bangelės).

1.3. Informacijos glaudinimo algoritmų poreikis

Vis labiau didėjanti kompiuterinės technikos ir internetinių technologijų sparta įtakojo, jog vaizdo, garso ir foto informaciją tapo lengvai pasiekiami, dažnai naudojama ir tuo pačiu būtina kompiuterijoje ir jos panaudojimo srityse. Nepaisant dabartinių kompiuterių pastoviosios atminties dydžių bei interneto ryšio spartos, daugeliu atveju neįmanoma ir nepageidaujama saugoti ar tuo labiau internetu platinti nesuglaudintos foto ar vaizdo medžiagos.

1.1 lentelė Skirtingos duomenų apimties siuntimo laikas el. paštu

Duomenų šaltinis	Dydis	Formatas	Siuntimo laikas
Foto nuotrauka	3,00 MB	BMP	12 sek.
	266 KB	PNG	1 sek.
	142 KB	JPG	< 1 sek.

1.1 lent. iliustruoja, kaip sutrumpėja laikas ir padidėja darbo sparta, siunčiant suglaudintus duomenis. Pirmas laikas BMP- nesuglaudinti duomenys (su pertekline informacija), antras laikas PNG užkoduoti duomenys be informacijos praradimo (be perteklinės informacijos) ir trečias laikas JPG užkoduoti duomenys, su informacijos praradimu (be perteklinės informacijos).

Fotoaparata, skanavimo įrenginį, grafikos redagavimo programą galima vadinti pirminiais statinio vaizdo (ne video) šaltiniais. Jie pateikia pradinę nesuglaudintą informaciją taškiniu vaizdu, sudarytu iš atskirų taškų, išdėstytų eilutėmis ir stulpeliais (dvimatis masyvas).

Nespalvotuose vaizduose, kiekvienas taškas aprašomas pilkumo lygiu. Tam gali būti skirti pvz. 8 bitai. Spalvoto vaizdo atveju vietoj pilkumo lygio naudojamas spalvos komponentės lygis. O spalvos komponentės yra 3- raudona, žalia, mėlyna (*angl. RGB*). Taigi, kiekvienas taškas aprašomas 3*8 bitų, t.y. 24 bitais. Šis formatas vadinamas „bitų žemėlapiu“ (*angl. Bitmap, BMP*).

Pavyzdys, atspindintis glaudinimo poreikį kompiuterijoje: tarkime 5 milijonų taškų fotoaparato nuotraukos, išsaugotos BMP formatu, apimtis $5000000 \cdot 3 \cdot 8 = 114\text{Mb} = 14\text{MB}$. Į 2GB atminties kortelę tilptų tik apie 136-ios tokios nuotraukos. Norint talpinti daugiau reikia suglaudinti šį vaizdą ir taip sumažinti jo apimtį. Dažnai dabartiniai vaizdo šaltiniai tai atlieka patys. Apžvelgsime, kaip tai atliekama.

1.4. Glaudinimo principai

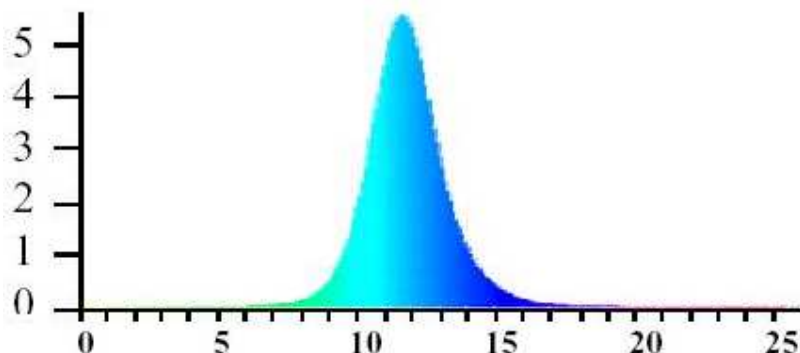
Vaizdų suglaudinimui pasitelkiama jų savybė- gretimi vaizdo taškai koreliuoja tarpusavyje, todėl yra perteklinės informacijos. Tokiu atveju, pagrindinė užduotis yra surasti mažiausiai besikoreliuojantį pradinio vaizdo atvaizdą. Dveji fundamentalaus glaudinimo principai yra pertekliško (*angl. redundancy*) ir nereikšmingumo (*angl. irrelevancy*) mažinimas [3]. Mažinant perteklišumą, siekiama pašalinti besidubliuojančias duomenų signalo (vaizdo) reikšmes. Nereikšmingumo mažinimo metu iš duomenų signalo yra pašalinamos tos reikšmės, kurios nebus pastebėtos signalo gavėjo. Kalbant apie vaizdus, tą įmanoma padaryti įvertinus žmogaus regos sistemos (*angl. human visual system*) savybes. Galima išskirti tokius pertekliško tipus:

- statistinis pertekliškumas
 - erdvinis pertekliškumas (*angl. spatial redundancy*)
 - kodavimo pertekliškumas
- vizualinis pertekliškumas (*angl. psychovisual redundancy*)
 - spalvinis pertekliškumas
 - dažninis pertekliškumas

Šių tipų pertekliško mažinimas gali žymiai įtakoti bendrą glaudinimo efektyvumą. Apžvelkime svarbiausiu iš jų.

1.4.1. Erdvinis pertekliškumas

Erdvinis pertekliškumas atsiranda todėl, kad pikselių reikšmės nėra tarpusavyje nepriklausomos. Paprastai yra labai didelė gretimų taškų koreliacija (1.2 pav.).



1.2 pav. Vaizdų gretimų taškų spalvų skirtumų histograma. X ašis- įvykio tikimybė, Y ašis- skirtumo reikšmė

Erdvinis pertekliškumas- tai statistinė autokoreliacija tarp vaizdo taškų [2]. Koreliacija – tai skaičius, kintantis nuo -1 iki 1 ir nusakantis dviejų dydžių tarpusavio ryšį. Tyrimais nustatyta, kad paprasto vaizdo autokoreliacija artima 0,98, o labai detaliuose vaizduose 0,45 – 0,75. Taigi su tikimybe, lygia autokoreliacijai galime teigti, kad kitas vaizdo taškas bus tos pačios spalvos, kaip ir esamas. Kadangi tikimybė (autokoreliacija) nėra lygi 1, ne visi spėjimai bus tikslūs – egzistuos spėjimo (numatymo) paklaida. Ją galima saugoti vietoje taško spalvos duomenų. Kadangi autokoreliacija didelė, spėjimo paklaidos kitimo intervalas paprastai būna daug mažesnis už galimų spalvos reikšmių intervalą, be to tik apie 50% vaizdo taškų spalva pakinta daugiau, nei 1 bitu. Tad spėjimo paklaidą galima koduoti mažesniu bitų kiekiu, pavyzdžiui 12, 16, nei pačią taško spalvą.

Detaliems vaizdams suglaudavimo laipsnis mažesnis, jį nusako Šenono (*orig. Claude Elwood Shannon, (1916–2001)*) teorema (žr. 1.8.1 sk.). Vienspalvio vaizdo atveju visų taškų spalva ta pati tad ir spėjimo paklaida lygi 0. Gausime duomenų seką, kurioje daug vienodų sekų. Šios sekos irgi turi perteklinės informacijos. Ją pašalinus, sumažės ir vaizdo apimtis. Tai kodavimo pertekliškumo mažinimas.

1.4.2. Kodavimo pertekliškumas

Kodavimo pertekliškumas atsiranda koduojant informacijos reikšmes į galutinį suglaudintą dvejetainį kodą.

1.2 lentelė Dvejetainio kodavimo pavyzdys

Simbolis	Pirmas kodas	Antras kodas	Tikimybė
S ₁	000	0000	0.1
S ₂	001	01	0.2
S ₃	010	1	0.5
S ₄	011	0001	0.05
S ₅	100	001	0.15

Lentelėje (1.2 lent.) pavaizduota abėcėlė iš 5 simbolių su kiekvieno simbolio pasirodymo tikimybe. Taip pat pateikti du unikalūs kodai, kurių pagalba galima koduoti duotosios abėcėlės simbolius. Naudojant pirmą kodą gaunamas toks vidutinis kodo dydis (2):

$$L_{avg,1} = 3 \text{ bitai} / 1 \text{ simboliui} ; \quad (2)$$

Naudojant antrą kodą, vidutinio kodo dydį galima rasti taip (3):

$$L_{avg,2} = 4 \times 0,1 + 2 \times 0,2 + 2 \times 0,5 + 4 \times 0,05 + 3 \times 0,15 = 1,95 \text{ bitų}. \quad (3)$$

Antrasis kodas taip pat unikalus, tačiau trumpesnis už pirmąjį, todėl galima daryti išvadą, jog sekoms koduoti efektyviau naudoti antrąjį kodą. Taip sumažinamas kodavimo pertekliškumas, neprarandant koduojamos informacijos.

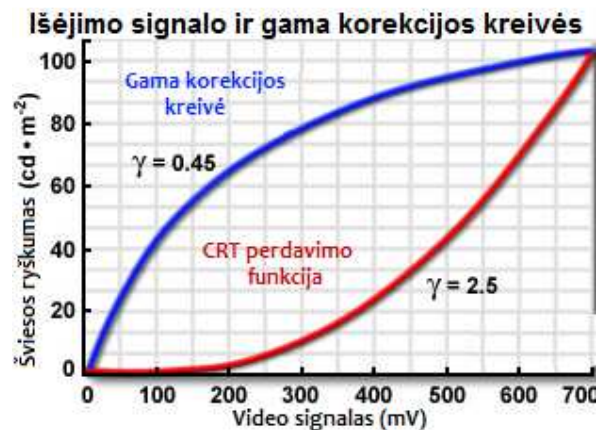
1.4.3. Spalvinis pertekliškumas

Spalvinio (intensyvumo) pertekliškumą nusako Vėberio (*angl. Weber–Fechner*) dėsnis, kuris teigia, jog pojūtis stiprėja proporcingai dirgiklio intensyvumo logaritmui. Arba paprasčiau: tarp stimulo intensyvumo (šiuo atveju spalvos) ir suvokiamo dydžio yra ne tiesinė, o funkcinė priklausomybė. Taigi pritaikius šį dėsnį, gautume, kad kuo stipresnis dirgiklis (intensyvesnė spalva), tuo daugiau jo stiprumas (spalvos intensyvumas) turi pasikeisti, kad pokytis būtų pastebėtas, o to pojūčio (spalvos intensyvumo) pasikeitimas atitinka logaritminę funkciją. Mažinti pertekliškumą pasitelkiama Gama korekcija (dar vadinama Gama glaudinimu). Tuo siekiama maksimaliai efektyviai išnaudoti ribotą bitų kiekį, skirtą informacijos saugojimui [7]. Akis šviesos intensyvumo didėjimą suvokia ne kaip tiesinę funkciją, bet kaip kėlimo laipsniu funkciją (gama funkciją) [8]. Formulėje (4) nurodoma, kaip gaunamas išvesties intensyvumas, turint įvesties intensyvumą, o A yra konstanta, paprastai lygi 1:

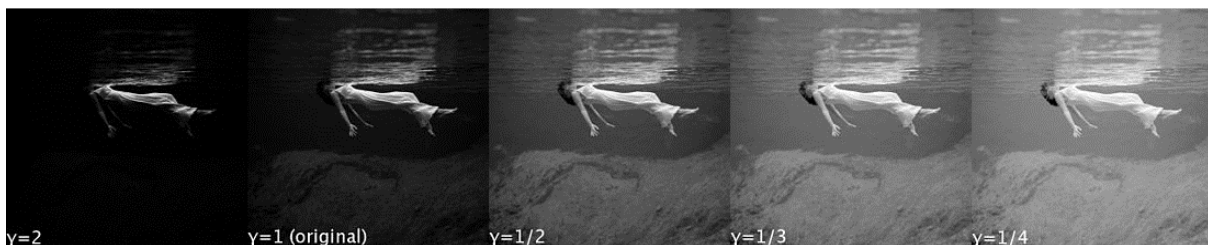
$$I_{I\check{s}} = A \times I_1^{\gamma} \quad (4)$$

čia γ – gama funkcijos reikšmė; $I_{I\check{s}}$ – išvesties intensyvumas; I_1 – įvesties intensyvumas.

Ekranų pikselio šviesumas nuo anodo įtampos priklauso ne tiesiškai, bet atitinka gama dėsnį [7]. Gama korekcijos pagalba atstatomas teisiškumas (žr. 1.3 pav.) 2.3 pav. Žmogaus šviesos intensyvumo interpretacija sutampa su CRT įvesties/išvesties charakteristikomis. Intensyvumą užkoduotą $1/(2,2) = 0.45$ gama reikšme kompensuoja automatinis CRT ekrano gama = 2,2 dekodavimas. Gama korekcija yra būtina visiems skaitmeniniams vaizdams, siekiant išsaugoti svarbią informaciją, nepriklausomai nuo vaizdo įrenginio [2].



1.3 pav. Gama korekcija. X ašis- Video signalo įtampa (mV). Y ašis- šviesos ryškumas (cd * m²)



1.4 pav. Gama korekcijos įtaka vaizdai (www.wikipedia.org)

Paveikslėlyje 1.4 pav. iliustruota gama korekcijos įtaka vaizdai. Jei gama funkcijos reikšmė γ (žr. formulę (4)) yra didesnė nei originalo ($\gamma = 1$), tuomet šešėliai tampa tamsesni. Ir atvirkščiai, kuo įtampa (1.3 pav. X ašis) mažesnė ($\gamma < 1$), tuo šešėliai šviesesni.

1.4.4. Dažninis pertekliškumas

Dažninis pertekliškumas taip pat atsiranda dėl žmogaus regos psichovizualinių savybių, vadinamų maskavimo efektais. Vaizdas, kuriame šie efektai pasireiškia, turi psichovizualinį informacijos perteklių. Žmogaus akis veikia kaip žemo dažnio filtras, todėl glaudinant vaizdą galima

panaikinti aukšto dažnio harmonikas ir tai bus nepastebima žmogaus akiai. Tai bene labiausiai išnaudojama savybė vaizdų glaudinimo algoritmuose.

Žmogaus akis pasižymi skirtingu jautrumu mėlynai, žaliai ir raudonai spalvoms. Mėlynai spalvai akis nejautriausia, žaliai- jautriausia. Atsižvelgiant į tai mėlyną spalvą galime koduoti mažiau tiksliai (skirdami mažiau bitų), o žalią spalvą koduoti tiksliau (skirti daugiau bitų). Kaip jau minėta, mažinant dažnio perteklišumą, turimų rezultatais, jog žmogaus akis mažiau jautri aukštadažnėms (dažnai kintančiomis) vaizdo komponentėms. Remdamiesi dažnio maskavimo efektu galime atmesti aukštadažnes vaizdo komponentes. Taip padidėja vaizdo taškų autokoreliacija. Autokoreliacija dar padidėja, kai spėjimui naudojamas ne vienas, o keli jau žinomi taškai iš tos pačios ir ankstesnių eilučių. Praktiškai užtenka 3 tokių taškų. Svarbu parinkti tinkamus svorio koeficientus, pritaikant prie koduojamo vaizdo.

1.4.5. Kodavimas

Šaltinyje [10] teigiama, jog kodas- tai taisyklių rinkinys, nusakantis kaip vienos formos informacija ar jos dalis yra konvertuojama į kitos formos informaciją ar jos dalį, nepriklausomai nuo to ar konvertuojama informacija yra to paties tipo (pvz.: skaitinė informacija konvertuojama į raidinę). Išskiriami tokie kodavimo tipai:

- Vaizdiniai. Vaizdinis kodavimas yra vaizdinių ir vaizdinės sensorinės informacijos kodavimas.
- Akustiniai. Akustinis kodavimas yra garsų, žodžių ir kitos girdimos informacijos kodavimas
- Semantiniai. Semantinis kodavimas yra sensorinių jutimų, kurie turi tam tikrą reikšmę arba gali būti panaudoti kontekste, kodavimas.

Išvardintų tipų kodavimai gali būti tiek su tiek ir be informacijos praradimo. Be jų gali būti naudojami ir kiti, tokie kaip entropinis duomenų kodavimas. Šio tipo kodavimas ir tam naudojami algoritmai plačiau bus apžvelgti 1.8, 1.9, 1.10 ir 1.11 skyriuose. Atsižvelgiant į šio darbo tikslą (žr. ĮVADAS), toliau bus nagrinėjami tik *iškraipantys* vaizdo glaudinimo algoritmai.

1.5. Standartinė vaizdo kodavimo schema

1.5 pav. pavaizduota schema, sudaryta iš trijų tarpusavyje susijusių komponentų- *vaizdo kodavimo, kvantavimo ir entropinio kodavimo* komponento [9]. Glaudinimo efektas pasiekiamas tokiu būdu- pirma pradinis vaizdas tiesiškai transformuojamas, kad sumažinti duomenų tarpusavio koreliaciją. Gauti transformacijos koeficientai yra kvantuojami ir, galiausiai, pritaikomas entropinis kodavimas.



1.5 pav. Standartinė vaizdo kodavimo schema

Vaizdo kodavimo (arba tiesinio transformavimo) algoritmų sukurta daug ir įvairių, kiekvienas turintis savo privalumų bei trūkumų. Galima paminėti tik kelis labiausiai paplitusius transformacijų algoritmus: DFT (*angl. Discrete Fourier Transform*) diskrečioji Furje transformacija, DCT (*angl. Discrete Cosine Transform*) diskrečioji kosinuso transformacija, DWT (*angl. Discrete Wavelet Transform*) diskrečioji bangelių transformacija [9].

Kvantavimo metu mažinamas saugojamų reikšmių tikslumas, taip sumažinama duomenų apimtis, reikalinga transformuotiems koeficientams saugoti. Šio proceso metu yra prarandama informacija (negrįžtamas procesas), nes tai yra atvaizdavimas daug-su-vienu (*angl. many-to-one mapping*). Kitvertus šio proceso metu yra pasiekama didžioji vaizdo kompresijos dalis. Kvantavimas gali būti atliekamas kiekvienai reikšmei atskirai (skaliarinis) arba atliekama tam tikrai koeficientų grupei (vektorinis).

Entropinio kodavimo metu toliau glaudinami kvantuotų reikšmių duomenys, naudojant glaudinimo algoritmus be informacijos praradimo, siekiant dar labiau sumažinti vaizdo failo apimtį. Labiausiai paplitę ir geriausi žinomi entropinio kodavimo algoritmai yra Hafmano, aritmetinio, RLE (*angl. Run-length coding*) bei žodyno kodavimas.

1.6. Glaudinimo algoritmų charakteristikos

Yra trys pagrindinės charakteristikos, kuriomis vertinami vaizdų glaudinimo algoritmai.

1.6.1. Glaudinimo lygis

Jį galima apibūdinti koeficientu α (žr. (5) formulę)(5), kuris parodo, kiek kartų suglaudintas vaizdas yra mažesnis už pradinį:

$$\alpha = \frac{I}{I'} ; \quad (5)$$

čia α – vaizdo glaudinimo koeficientas, I – pradinio vaizdo dydis (baitais), I' – suglaudinto vaizdo dydis (baitais).

Glaudinimo koeficientas parodo, kiek kartų suglaudintas vaizdas yra mažesnis už pradinį vaizdą. Paprastai šis dydis labai įtakoja vaizdo kokybę- kuo aukštesnis šis koeficientas, tuo prastesnė suglaudinto vaizdo kokybė [9]. Šį faktą labai svarbu įvertinti, glaudinant ar lyginant kelis suglaudintus vaizdus. Be to, naudojant tam tikrus algoritmus, glaudinimo lygis gali būti labai priklausomas nuo paties vaizdo. Vaizdų glaudinimo srityje šis aspektas vadinamas duomenų priklausomumu (*angl. data dependency*). Naudojant algoritmą, kuris pasižymi dideliu duomenų priklausomumu, glaudinant didelio detalumo vaizdą (pvz.: religinės šventės žmonių sambūris), gali būti pasiekiamas labai nedidelis glaudinimo lygis. Tačiau vaizdui, kuris nepasižymi detalumu (pvz., vienspalvės sienos), tas pats algoritmas gali pasiekti labai didelį glaudinimo lygį.

1.6.2. Glaudinimo greitis

Laikas, kuris sunaudojamas vaizdo suglaudinimui bei atstatymui, vadinamas glaudinimo greičiu. Jis labiausiai priklauso nuo glaudinimo algoritmo sudėtingumo, efektyvaus algoritmo realizavimo bei nuo techninės įrangos greičio ir architektūros.

Nors tam tikrais atvejais glaudinimo greitis yra kritinė charakteristika, daugeliu atveju labiausiai vertinamos yra glaudinimo lygio ir vaizdo kokybės charakteristikos.

1.6.3. Glaudinto vaizdo kokybė

Suglaudinto ir dekoduo to vaizdo kokybė apibrėžia tikslumą, kuriuo yra atkuriamas vaizdas iš suglaudintų duomenų, palyginant su pradiniu (nesuglaudintu) vaizdu.

Suspaudimo algoritmai, naudojantys vaizdo suspaudimo be informacijos praradimo schemas, išlaiko visą originalią informaciją. Tad ši charakteristika taikoma tik algoritmams, kurie naudoja suspaudimo schemas su informacijos praradimu. Vaizdo kokybei įvertinti dažniausiai naudojama vidutinė kvadratinė paklaida, tačiau tai nėra vienintelis įmanomas kokybės (paklaidos) įvertis. Vidutinė kvadratinė paklaida randama pagal (6) formulę.

$$MSE = \frac{1}{M \times N} \sum_{y=1}^M \sum_{x=1}^N [I(x, y) - I'(x, y)]^2 ; \quad (6)$$

čia MSE – vidutinė kvadratinė paklaida, $M \times N$ – vaizdo dimensijos (aukštis ir plotis) pikseliais, $I(x, y)$ – originalaus vaizdo konkretaus taško reikšmė, $I'(x, y)$ – suspausto vaizdo konkretaus taško reikšmė.

Kuo labiau suspausto vaizdo taškų reikšmės skiriasi nuo originalaus vaizdo taškų reikšmių, tuo didesnį yra vidutinė kvadratinė paklaida. Jei keli algoritmai spaudžia vienodą vaizdą, tai geriausių vaizdo kokybę atkuria algoritmas, kurio MSE įvertis yra mažiausias.

1.7. Plačiausiai paplitę iškraipantys vaizdų glaudinimo metodai

Darbe naudojamas algoritmas, kaip duomenų sekų šaltinis, yra JPEG. Tai populiariausias, greičiausias ir labai efektyvus vaizdų glaudinimo algoritmas. Tačiau tam, kad galėtume palyginti jo privalumus ar trūkumus, su kitais, panašaus tipo algoritmais, todėl šiame skyriuje apžvelgsime ir BTC. JPEG ir BTC yra iškraipantys vaizdų glaudinimo algoritmai.

1.7.1. BTC algoritmas

BTC (*angl. Block Truncation Coding*) algoritmas yra vienas iš vaizdų glaudinimo algoritmų, naudojančių vaizdo glaudinimo su informacijos praradimu schemą [13]. Algoritmo idėja – suskaidyti vaizdą į nedidelius blokus ir kiekvienam blokui sumažinti jo pilkumo skalės lygius. Kvantavimas atliekamas atsižvelgiant į vaizdo bloko reikšmių statistiką ir pasiskirstymus. Nauji pilkumo lygiai parenkami taip, kad būtų minimizuotas pasirinktas klaidos kriterijus, o tada visos bloko reikšmės pakeičiamos dviem naujais lygiais [9].

Baziniame algoritmo variante vaizdas skaidomas 4×4 pikselių vaizdo blokus. Bloko pikselių reikšmės koduojamos dviem lygiais – a ir b . Kvantavimo lygiai a ir b parenkami taip, kad būtų išsaugomi pirmieji momentai v_1 ir v_2 (7):

$$v_1 = \frac{1}{m} \sum_{i=1}^m x_i ; v_2 = \frac{1}{m} \sum_{i=1}^m x_i^2 ; \quad (7)$$

čia m yra vaizdo bloko (fragmento) pikselių kiekis, x_i – vaizdo bloko i - tasis pikselis.

Norint suskaičiuoti kvantavimo lygių a ir b reikšmes, sprendžiama lygčių sistema (8):

$$\begin{cases} \frac{1}{m} ((m - q) \times a + q \times b) = v_1 \\ \frac{1}{m} ((m - q) \times a^2 + q \times b^2) = v_2 \end{cases} ; \quad (8)$$

čia m yra vaizdo bloko (fragmento) pikselių kiekis, q – pikselių, kuriems bus priskirta reikšmė b , skaičius; a ir b – nauji bloko kvantavimo lygiai; v_1 ir v_2 – bloko pirmos ir antros eilės momentai.

Reikšmė q randama tokiu būdu: i - tajam pikseliui priskiriama reikšmė a , jeigu $x < x_{slenkstinė}$, kitu atveju priskiriama reikšmė b . Baziniame algoritmo variante $x_{slenkstinė} = v_1$.

Išsprendus lygčių sistemą (9), randami kvantavimo lygiai:

$$a = v_1 - \delta \sqrt{\frac{q}{m - q}}, b = v_1 - \delta \sqrt{\frac{m - q}{q}} ; \quad (9)$$

čia a ir b – nauji bloko kvantavimo lygiai, v_1 – bloko pirmos eilės momentas, q – pikselių, kuriems bus priskirta reikšmė b , skaičius, m – vaizdo bloko (fragmento) pikselių kiekis. Dydis δ , randamas pagal (10) formulę:

$$\delta = \sqrt{v_2 - v_1^2} ; \quad (10)$$

Suglaudintam blokui užkoduoti reikia 3 dydžių- a , b ir B . Čia B -bitinė plokštuma, 1-am pikseliui priskirta a reikšmė, 2- am pikseliui priskirta b reikšmė. Glaudinimo koeficientą galima rasti pagal (11) formulę:

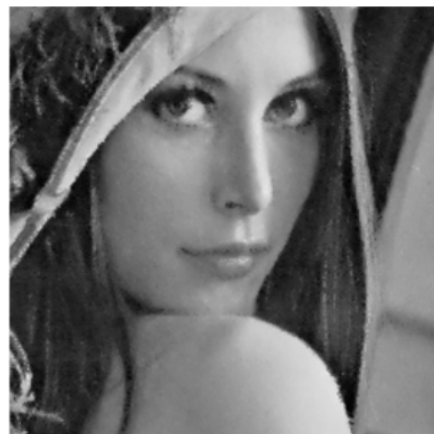
$$\alpha = \frac{m \times p}{2 \times p + m}; \quad (11)$$

čia a - vaizdo glaudinimo koeficientas, m - vaizdo bloko pikselių kiekis, p - bitų kiekis, skiriamas vienam pikseliui koduoti (paprastai $p = 8$).

Žemiau pateikti nesuglaudinto (1.6 pav.) ir BTC algoritmu glaudinto paveikslėlių (1.7 pav.) pavyzdžiai.



1.6 pav. Nesuglaudintas dvispalvis paveikslėlis
(www.opticalengineering.spiedigitallibrary.org)



1.7 pav. Suglaudintas BTC algoritmu dvispalvis
paveikslėlis
(www.opticalengineering.spiedigitallibrary.org)

BTC algoritmo privalumai:

- labai greitai veikiantis ir mažai skaičiavimų reikalaujantis algoritmas;
- lengvai suprantamas ir realizuojamas;
- esant nedideliame glaudinimo koeficientui išlaikoma labai gera atkurto vaizdo kokybė.

BTC algoritmo trūkumai:

- pasiekiamas nedidelis vaizdo glaudinimo koeficientas, paprastai neviršijantis 4-8 kartų;
- naudojant didesnių glaudinimo lygį, pasireiškia blokinė struktūra ir laiptinis spalvų efektas.

BTC ir JPEG glaudinimo principai šiek tiek panašūs- vaizdas skaidomas į blokus, tačiau yra ir skirtumų- BTC nenaudoja diskrečiosios kosinuso transformacijos. BTC sumažinami spalvų lygiai, o JPEG spalvų dedamųjų elementai yra skaičiuojami nuo iš kairės esančio elemento. Tobulinant BTC algoritmą, yra sukurta daug jo modifikacijų. Yra modifikacijų ir spalvotiems vaizdams glaudinti, tačiau vis tiek nėra pasiekiami glaudinimo lygiai, artimi DCT pagrindu sukurtiems glaudinimo algoritmams. Todėl išsamiau nagrinėsime DCT naudojančią JPEG algoritmą.

1.7.2. JPEG kodavimo algoritmas

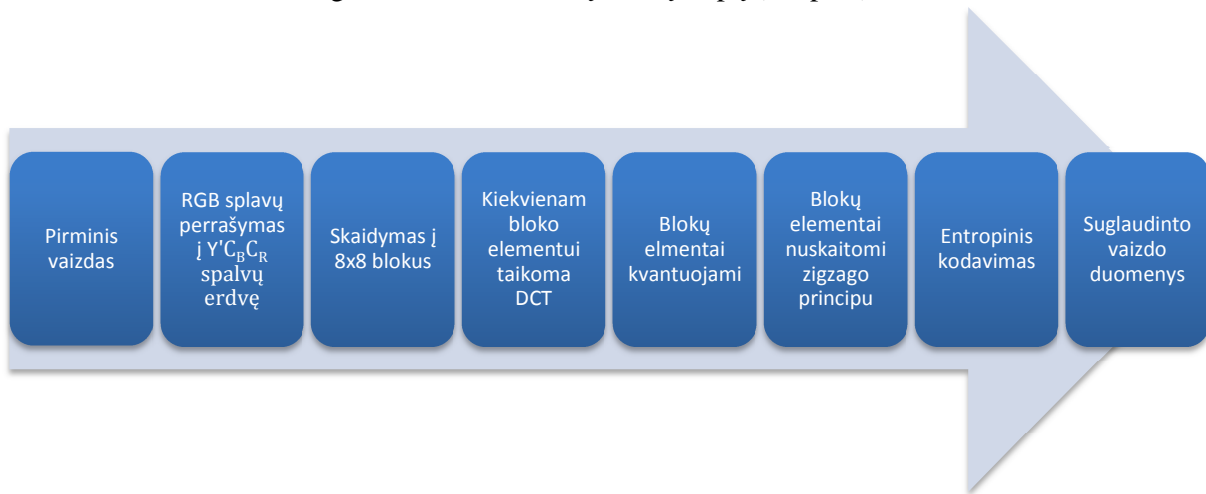
1992 metais jungtinė fotografų ekspertų grupė (*angl. JPEG*) išleido vaizdų glaudinimo standartą JPEG [4]. Jame aprašyta vaizdų kodavimo (glaudinimo) algoritmas, bei failo, kuriame saugomi suglaudinti duomenys, formatą. Bendrine prasme JPEG yra kodekas, apibrėžiantis, kaip 2D statinis vaizdas transformuojamas į bitų srautą ir atgal į vaizdą, tačiau nenurodo failo struktūros. Failo formatą apibrėžia EXIF ir JFIF standartai [2].

Numatyti keturi pagrindiniai suspaudimo tipai [9]: nuoseklus, progresyvinis ir hierarchinis (kodavimui su informacijos praradimu), bei vienas režimas be informacijos praradimo. Labiausiai paplitęs yra nuoseklus suspaudimo algoritmas.

JPEG patariama glaudinti realistiškus (fotonuotrauka, skanuotas piešinys) vaizdus, kuriuose dominuoja tolygūs spalvų ir fonų pokyčiai. Brėžiniams, tekstiniams vaizdams, grafikams JPEG formatas netinkamas. Dėl informacijos praradimo glaudinant vaizdą, šis formatas netinkamas ir vaizdų saugojimui tarp redagavimo etapų. Taigi, jei reikia redaguoti fotonuotrauką, geriau ją fotoaparate saugoti RAW ar panašiu formatu.

1.7.2.1. JPEG kodavimo algoritmo žingsniai

JPEG kodavimo algoritmas, skirstomas į keletą etapų (1.8 pav.):



1.8 pav. JPEG kodavimo algoritmas

1. Spalvų erdvės keitimas

Informacija iš RGB erdvės atvaizduojama į $Y'CbCr$ spalvų erdvę. R'G'B' taško atvaizdavimas $Y'CbCr$ erdvėje, apskaičiuojamas taikant (12), (13)(8) ir (14) formules [5] :

$$Y = (0,299 * R') + (0,587 * G') + (0,114 * B') \quad (12)$$

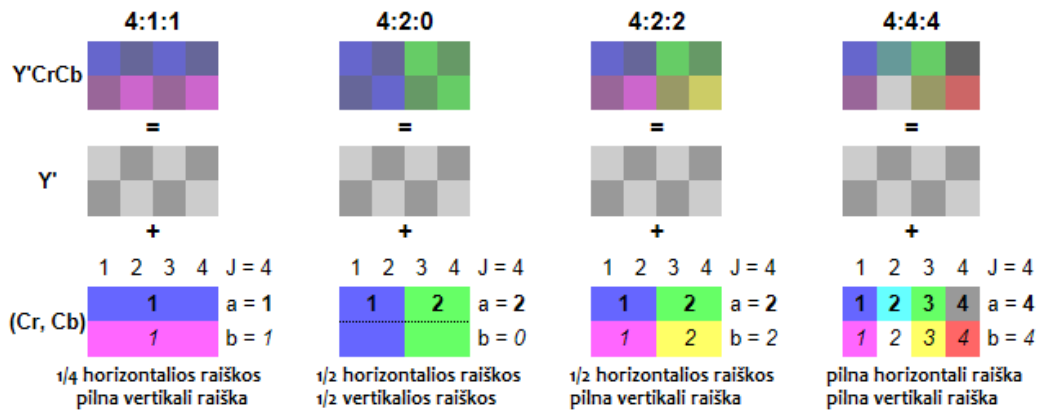
$$C_B = 128 - (0,1687 * R') - (0,3313 * G') + (0,5 * B') \quad (13)$$

$$C_R = 128 + (0,5 * R') + (0,4187 * G') + (0,0813 * B') \quad (14)$$

čia Y' - šviesumas (luma) su gama korekcija, C_B ir C_R – chroma spalvų komponentės. Į $Y'CbCr$ spalvų erdvę nekonvertuojama, jei reikia maksimalios vaizdo kokybės.

2. Chromatiškumo raiškos mažinimas

Žmogaus akis yra mažiau jautri atspalvio nei intensyvumo pokyčiui [6]. JPEG standarte numatyta galimybė taikyti chromatiškumo raiškos mažinimą (*angl. chroma downsampling/ subsampling*). Raiškos mažinimo schema paprastai žymima 3 skaičiais, atskirtais dvitaškiu (pvz.: 4:2:2) [6], nors galimas ir žymėjimas keturiais skaičiais (pvz.: 4:2:2:4).



1.9 pav. Chromatiškumo raiškos mažinimo schemas ir koeficientai

Dėl saugojimo ir perdavimo apribojimų, visada yra tikslas suglaudinti duomenis, o dėl paminėtos žmogaus regos specifikos, sumažinus chromatiškumo raišką trečdaliu (pvz.: 4:2:2 Y'CrCb vietoje 4:4:4 R'G'B'), tai vaizdą stebinčiam žmogui liks nepastebėta.

Pateiktose schemose (1.9 pav.) tik konceptualiai iliustruojami galimi chromatiškumo raiškos mažinimo variantai. Kaip jau minėta, schema žymima trimis skaičiais, pirmasis iš jų J - tai konceptualaus vaizdo elementų bloko plotis pikseliais (paprastai lygus 4-iems), a - pirmoji chromatiškumo komponenčių (C_B ir C_R) eilutė, b - antroji (papildoma) chromatiškumo komponenčių (C_B ir C_R) eilutė. a ir b eilutės gali būti dalomos į daugiausiai keturias dalis (chromatiškumo raiška nemažinama) arba nedalomos (chromatiškumo raiška sumažinama $\frac{3}{4}$).

JPEG standarte naudojama 4:2:0 schema, taip sumažinant vertikalią ir horizontalią raiškas perpus t.y. Y' komponentei saugoti naudojami 8 bitai, tuo tarpu C_B ir C_R komponentėms saugoti skiriama po 4 bitus. Bendras duomenų kiekis, reikalingas išsaugoti vieno pikselio spalvinei informacijai sumažinamas nuo 3-jų iki 2-jų baitų.

3. Skaidymas į blokus

Kiekviena Y'CrCb (arba R'G'B') komponentė skaidoma į 8x8 blokus. Kiekvienas blokas koduojamas atskirai. Bloko dydis parinktas pakankamai didelis, dažnio komponentams užfiksuoti, bet ne per didelis, vengiant globalių kodavimo artefaktų [2],[4]. Jei matricos elementų nepakanka 8x8 blokui užpildyti, pridedamos papildomos reikšmės- juodi ar vaizdo kraštą replikuojantys pikseliai. Taip gaunami (užpildomi) vaizdo duomenų 8x8 blokai.

4. DCT skaičiavimas

Kiekvienam blokui taikoma DCT transformacija [4]. DCT (*angl. Discrete Cosine Transform*)- Diskrečioji kosinuso transformacija, kuri išreiškia baigtinę taškų seką funkcijų kosinusu suma, o tos funkcijos keičiasi skirtingais dažniais [11]. Šiame etape suspaudimas nepasiekiamas, o tiesiog duomenų erdvė pakeičiama į dažnių erdvę. Skaičiuojant DCT visam vaizdai, reikia didelių skaičiavimo resursų. Norint to išvengti DCT skaičiuojama kiekvienam vaizdo blokui atskirai [4]. Visų blokelių DCT suskaičiavimui atskirai, reikia žymiai mažiau resursų. Tačiau šis vaizdo skaidymas blokeliais yra pagrindinis JPEG algoritmo trūkumas- prastesnės kokybės vaizde blokelių ribos išsiskiria, yra aiškiai matomos.

Kaip pavyzdį, galime naudoti improvizuoto 8x8 paveikslėlio vaizdo duomenų bloką (1.10 pav.).

$$\begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{bmatrix}.$$

1.10 pav. Vaizdo duomenų blokas (www.wikipedia.org)

Prieš naudojant DCT, bloko reikšmės iš $[0; 255]$ perslenkamos į $[-128; 127]$ - įvykdoma sąlyga DCT skaičiavimui. Taip reikšmių pasiskirstymo centru tampa 0-ė reikšmė, kaip koordinatinių sistemos centras (1.11 pav.) :

$$g = \begin{matrix} & & & & x & & & & \\ & & & & \longrightarrow & & & & \\ \begin{matrix} -76 & -73 & -67 & -62 & -58 & -67 & -64 & -55 \\ -65 & -69 & -73 & -38 & -19 & -43 & -59 & -56 \\ -66 & -69 & -60 & -15 & 16 & -24 & -62 & -55 \\ -65 & -70 & -57 & -6 & 26 & -22 & -58 & -59 \\ -61 & -67 & -60 & -24 & -2 & -40 & -60 & -58 \\ -49 & -63 & -68 & -58 & -51 & -60 & -70 & -53 \\ -43 & -57 & -64 & -69 & -73 & -67 & -63 & -45 \\ -41 & -49 & -59 & -60 & -63 & -52 & -50 & -34 \end{matrix} & & & & & & & & y. \\ & & & & & & & & \downarrow \end{matrix}$$

1.11 pav. Vaizdo duomenų bloko reikšmės po perslinkimo (www.wikipedia.org)

Po perslinkimo, blokams skaičiuojama DCT. JPEG glaudinime naudojama DCT forma, pateikta (15) formulėje:

$$G_{u,v} = \sum_{x=0}^7 \sum_{y=0}^7 \alpha(u)\alpha(v)g_{x,y} \cos \left[\frac{\pi}{8} \left(x + \frac{1}{2} \right) u \right] \cos \left[\frac{\pi}{8} \left(y + \frac{1}{2} \right) v \right] \quad (15)$$

čia u - horizontalus erdvinis dažnis (eilutės numeris), sveikasis skaičius nuo $0 \leq u < 8$, v - horizontalus erdvinis dažnis (stulpelio numeris), sveikasis skaičius nuo $0 \leq v < 8$.

DCT skaičiavimui būtina sąlyga- matricos ortogonalumas (statmenumo sąvokos apibendrinimas bet kokios tiesinės erdvės su apibrėžta skaliarine sandauga elementams), pasiekama dėl α koeficientų, kurie užtikrina, kad matrica bus ortogonalinė [11]:

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{8}}, & \text{jei } u = 0 \\ \sqrt{\frac{2}{8}}, & \text{kitu atveju} \end{cases} \quad \alpha(u) - \text{tai normalizacijos skalės koeficientas, užtikrinantis, kad matrica bus ortogonalinė.}$$

čia $g_{x,y}$ - vaizdo elemento reikšmė koordinatėse (x, y) , $G_{u,v}$ - DCT koeficiento reikšmė koordinatėse (u, v) .

$$G = \begin{matrix} & & & & u & & & & \\ & & & & \longrightarrow & & & & \\ \begin{matrix} -415.38 & -30.19 & -61.20 & 27.24 & 56.13 & -20.10 & -2.39 & 0.46 \\ 4.47 & -21.86 & -60.76 & 10.25 & 13.15 & -7.09 & -8.54 & 4.88 \\ -46.83 & 7.37 & 77.13 & -24.56 & -28.91 & 9.93 & 5.42 & -5.65 \\ -48.53 & 12.07 & 34.10 & -14.76 & -10.24 & 6.30 & 1.83 & 1.95 \\ 12.12 & -6.55 & -13.20 & -3.95 & -1.88 & 1.75 & -2.79 & 3.14 \\ -7.73 & 2.91 & 2.38 & -5.94 & -2.38 & 0.94 & 4.30 & 1.85 \\ -1.03 & 0.18 & 0.42 & -2.42 & -0.88 & -3.02 & 4.12 & -0.66 \\ -0.17 & 0.14 & -1.07 & -4.19 & -1.17 & -0.10 & 0.50 & 1.68 \end{matrix} & & & & & & & & v. \\ & & & & & & & & \downarrow \end{matrix}$$

1.12 pav. DCT koeficientų blokas (www.wikipedia.org)

Po DCT skaičiavimo gautas 8x8 dažninių koeficientų blokas, pateiktas **1.12 pav.** Šio bloko kairysis viršutinis koeficientas yra didžiausias (dėl DCT specifikos kaupti didžiąją informacijos dalį viename kampe) [11]. Jis vadinamas DC koeficientu, o likę 63 elementai yra AC koeficientai. DC koeficiento reikšmė koduojama nepakitusi, tuo tarpu vietoje visų AC koeficientų reikšmių yra koduojamos tik skirtumas nuo prieš tai buvusio elemento (iš kairės į dešinę per visas eilutes) [4]. Kaip ir Furje transformacija, DCT pateikia dvimatį dažnio spektrą, tačiau labiau, nei Furje, koncentruoja energiją žemųjų dažnių srityje [2]. Taip labiau išnaudojamas dažnio maskavimo efektas.

Kaip matome DCT koeficientų bloko (1.12 pav.) elementų reikšmės yra didesnės nei galima užkoduoti vienu baitu (reikšmės intervale nuo -128 iki 127), todėl tam tikru DCT skaičiavimo metu, šioms reikšmėms saugoti gali būti naudojama 11 ir daugiau bitų. Atminties dydis, naudojamas reikšmėms saugoti, atstatomas iki vieno baito sekančiame kvantavimo etape.

5. Kvantavimas

DCT koeficientai kvantuojami. Kvantavimo metu 8x8 bloko elementų reikšmės yra apvalinamos naudojant kvantavimo lenteles- tai 8x8 dydžio lentelė su joje surašytais koeficientais [4]. Priklausomai nuo vartotojo pasirinktos norimos vaizdo kokybės, yra parenkamos dvi kvantavimo lentelės- po atskirą lentelę skaisčio (Y') ir chromatiškumo (C_B ir C_R) reikšmėms. Kuo koeficientas didesnis, tuo labiau sumažinama vaizdo bloko elemento reikšmė (dalyba iš koeficiento). Y' (Luma) komponentės lentelėje koeficientai mažiausi (1.13 pav.), chromatiškumo (C_B ir C_R) komponentių lentelėje- didesni (1.14 pav.).

$$Q_l = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

1.13 pav. Tipinė (50 %) skaisčio (komponentės Y') kvantavimo matrica (www.ams.org).

$$Q_c = \begin{bmatrix} 17 & 18 & 24 & 47 & 99 & 99 & 99 & 99 \\ 18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\ 24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\ 47 & 66 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{bmatrix}$$

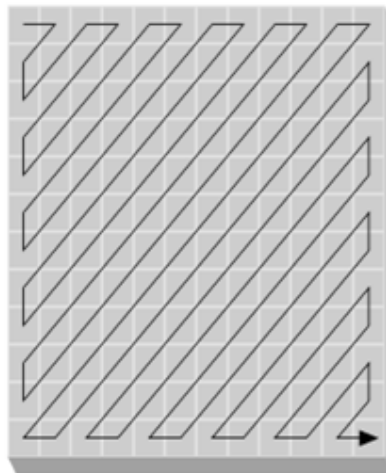
1.14 pav. Tipinė (50 %) chromatiškumo (komponenčių C_B ir C_R) kvantavimo matrica (www.ams.org).

Iš C_R ir C_B komponenčių paliekamos tik žemadažnės dedamosios. Taip išnaudojamas dažnio maskavimo efektas. Koefficientų reikšmės paskaičiuojamos pagal pasirinktą glaudinimo laipsnį, dažniausiai išreiškiamą procentais. Tokio kvantavimo trūkumas- svarbi aukštadažnė dedamoji bus ignoruojama, dėl to detalus vaizdas taps kiek išsiliejęs.

6. Zigziginis nuskaitymas

Koefficientų matrica nuskaityta zigzago principu ir užkoduojama RLE, Hafmano ar kitu aritmetiniu kodu. Tai yra antrasis suspaudimo algoritmas (bet be informacijos praradimo), kuris pritaikomas realizuojant JPEG vaizdų suspaudimo standartą.

Kvantuotuose (suapvalintuose) 8×8 blokuose didžioji dalis elementų lygi 0. Kuo aukštesnis dažnis (kuo toliau nuo pirmosios bloko reikšmės, reiškiančios nuolatinę dedamąją, tuo šių 0 daugiau. Efektyviam kodavimui reikia, kad šie 0 sudarytų kuo ilgesnę seką. Tai vadinama entropijos kodavimu. Po jo tarp elementų neturi likti tarpusavio koreliacijos. JPEG algoritme tam naudojamas zigziginis skanavimas (1.15 pav.)– jis 8×8 bloką (dvimatį masyvą) paverčia 64 elementų eilute (vienmačiu masyvu), kurioje elementai išrikiuoti dažnio didėjimo kryptimi.



1.15 pav. Sekos iš matricos sudarymas zigzago principu.

7. Entropinis glaudinimas

Po zigziginio nuskaitymo gauta ir surikiuota eilutė gali būti efektyviai užkoduota. Kodavimo pertekliškumui sumažinti naudojamas RLE glaudinimas. Po RLE glaudinimo gautos sekos papildomai gali būti koduojamos Hafmano algoritmu- visos galimos eilutės elementų reikšmės surikiuojamos pagal tikimybę. Kuo didesnė tam tikros reikšmės pasirodymo tikimybė, tuo mažesnė informaciją ji neša ir tuo trumpesnė bitų seka jai koduoti turi būti priskirta. Kuo tikimybė mažesnė, tuo bitų seka ilgesnė. Gaunama bitų seka, atitinkanti pradinį vaizdą su informacijos praradimu. Ši seka siunčiama komunikacijų sistemomis arba išsaugoma laikmenoje pridedant specialių simbolių, žyminčių bloko, eilutės pabaigą. Visoms vaizdo kodavimo operacijoms, išskyrus kvantavimą, yra

atvirkštinės operacijos. Jos taikomos atkuriant vaizdą- pirmiausiai vyksta dekodavimas, tuomet atvirkščiasis zigzaginis skanavimas, atvirkštinė diskrečioji kosinuso transformacija, spalvos perrašomos (jei reikia) iš $Y'CbCr$ į $R'G'B'$ erdvę, blokai atstatomi eiliškumo tvarka. Gaunamas pradinis vaizdas be dalies informacijos, kuri buvo prarasta kvantavimo metu. JPEG formatu suglaudintą vaizdą galima sukroti 90 laipsnių kampu, apversti pagal horizontalią, vertikalią ašį, įstrižainę neprarandant kokybės. Šių operacijų metu tik sukeičiama 8x8 vaizdo blokų išdėliojimo tvarka ir kryptis ir neatliekama kvantavimo operacija, sąlygojanti informacijos praradimą. Paprastai grafikos redaktoriuose šios operacijos pažymimos, kad yra be informacijos praradimo (*angl. lossless*).

Toliau pateikiama paveikslėliai, gauti panaudojus skirtingą glaudinimo laipsnį (pritaikytos skirtingos kvantavimo matricos). Didėjant glaudinimo laipsniui, ryškėja algoritme naudojamo skaidymo į 8x8 blokelius pasekmės.



1.16 pav. 100 proc. kokybės suglaudintas paveikslėlis (www.wikipedia.org)

100 proc. kokybės JPEG vaizde (1.16 pav.) sunku įžiūrėti skirtumą tarp nesuglaudinto (TIFF ar RAW), kurie užimtų maždaug 6 kartus daugiau vietos.



1.17 pav. 60 proc. kokybės JPEG suglaudintas paveikslėlis

Tokios kokybės paveikslėlyje (1.17 pav.) tik įdėmiai įsižiūrėjus galima pastebėti informacijos praradimo įtaką. Tokio lygio suglaudimas dažniausiai naudojamas interneto svetainėse ir el. pašte.



1.18 pav. 10 proc. kokybės JPEG suglaudintas paveikslėlis

1.18 pav. labai ryškiai matomas informacijos netekimas bei suskaidymas į blokelius koduojant.

1.7.2.2. JPEG failo struktūra

JPEG standarte nenurodyta failo struktūra, o jai aprašyti naudojami EXIF ir JFIF standartai [2]. Remiantis šiais standartais, JPEG failas sudarytas iš segmentų, kuriuos identifikuoja žymekliai. Be koduotų duomenų, papildomai gali būti pateikiamos Hafmano lentelės, kvantavimo lentelės bei kiti kodavimo parametrai ir papildoma su vaizdu susijusi informacija.

Žymekliui skiriami du baitai, pirmasis baitas visuomet žymimas šešioliktainiais skaičiais „FF“. Po žymeklio (išskyrus failo pradžios ir pabaigos žymeklius) fiksuojamas 00 baitas, po kurio eina reikšmė, nurodanti segmento dydį.

JPEG failo struktūroje naudojami žymekliai [2]:

- FF D8: Failo pradžia
- FF E0: Programos segmentas 0
- FF E1: Programos segmentas 1
- FF C0: Kadro pradžia
- FF C4: Hafmano lentelės pradžia
- FF DB: Kvantavimo lentelės pradžia
- FF DA: Duomenų segmento pradžia
- FF D9: Failo pabaiga

Paveikslėlyje 1.19 pav. pateikiamas JPEG failo struktūros fragmentas.

000000	FF	D8	FF	E0	00	10	4A	46	49	46	00	01	01	00	00	01
000010	00	01	00	00	FF	FE	00	58	43	52	45	41	54	4F	52	3A
000020	20	58	56	20	56	65	72	73	69	6F	6E	20	33	2E	31	...

1.19 pav. JPEG failo fragmentas

1.7.2.3. JPEG privalumai ir trūkumai

Privalumai:

- Populiariausias iš šiuo metu naudojamų vaizdų glaudinimo algoritmų;
- Tinka dinaminio vaizdo kadrams koduoti;
- Pradinis failas gali būti sumažinamas iki 10 ar net 20 kartų su vizualiai nepastebimais pokyčiais.

Trūkumai:

- Netinka 1 baito ir mažiau spalvų vaizdams glaudinti;
- Nepakankamai efektyviai glaudina brėžinių, tekstinės ar pieštos grafikos vaizdą;
- Esant dideliame suspaudimo laipsniui išryškėja blokinė struktūra;

- Pasireiškia taip vadinamas kontūrų aidas.

1.7.2.4. JPEG duomenų sekų specifika

Dėl keleto JPEG savybių, šis algoritmas yra specifinis jo generuojamų duomenų sekų atžvilgiu. Tokia JPEG standarto specifika aktuali, siekiant šiame darbe užsibrėžto tikslo. Kaip minėta anksčiau, JPEG standarte naudojama DCT (žr. 4 DCT skaičiavimas) statistiškai didžiąją dalį signalo (šaltinio) informacijos kaupia viename kampe (JPEG atveju tai karys viršutinis kampas). Kvantavimo (žr. 5 Kvantavimas) metu yra pašalinamos aukštadažnės dedamosios, jų reikšmės suapvalinant iki artimiausio sveikąjį skaičių, kuris esant didesniems vaizdo suspaudimo laipsniams dažniausiai yra „0“. Galima teigti, kad kuo didesnis suspaudimo laipsnis (kvantavimo matricos elementai didesni), tuo daugiau sekoje generuojama nulių. Dėl specifinio bloko nuskaitymo zigzago (žr. 6 Zigzaginys nuskaitymas) principu, taip sustatytos sekos paskiausi iš eilės einantys elementai yra nuliai- sudaroma vientisa skaičiaus „0“ eilė. Zigzago nuskaitymo principo panaudojamas, grindžiamas tuo, kad gauta seka būtų kuo efektyviau užkoduojama. Tam naudojami entropinio kodavimo algoritmai. Apžvelkime populiariausius iš jų, taip pat jų efektyvumą nusakančius kriterijus.

1.8. Duomenų entropija

Duomenys sudaro tam tikras sekas, kurios kompiuterijoje koduojamos dvejetainiu (binariniu) kodu [10]. Toks kodavimas gali turėti kodavimo perteklišumą (žr. 1.4.2 posk.). Schemose, apibrėžiančiose tokių sekų kodavimą, naudojami algoritmai, vadinami entropiniais (*angl. entropy*). Pastarasis terminas yra perimtas iš termodinamikos, kur entropija naudojama apibrėžti sistemos netvarkingumo laipsnį [14]. Taigi šiuo atveju galima laikyti, jog entropija informatikoje- tai *duomenų sistemos* netvarkingumo matas. Norint įvertinti ir apibrėžti entropinius glaudinimo algoritmus, yra naudojami keletas jų efektyvumą nusakančių kriterijų. Vienas iš jų tai šaltinio entropija. Ją nusako Šenono teorema- šaltinio duomenų entropija tai yra bitų (dvejetainių skaitmenų) kiekis reikalingas tiems duomenims užkoduoti. Ši entropija nieko nesako apie pačios informacijos vertę.

Konceptualiai entropija termodinamikoje ir informatikoje yra tapačios. Tačiau yra ir svarbių skirtumų. Pirmiausia, termodinamikoje entropija yra išreiškiama energijos matavimo vienetais priklausomai nuo temperatūros, o Šenono entropija matuojama bitais, kurie neturi išmatavimų. Šis skirtumas tėra susitarimo reikalas. Panagrinėkime, kaip atsirado Šenono teorema.

1.8.1. Šenono entropija ir teorema

Šenonas (*orig. Claude Elwood Shannon, (1916–2001)*), pritaikydamas fiziko Bolcmano (*vok. Ludwig Eduard Boltzmann, (1844 – 1906)*) teoremą informatikos teorijoje, atsitiktinio diskrečiojo dydžio X (su galimomis reikšmėmis intervale $\{x_1, \dots, x_n\}$) entropiją pažymėjo raide H , o šio dydžio pasirodymo tikimybę $P(X)$ [15]. Tuomet šiuos dydžius sustačius į (16) formulę, gauname:

$$H(X) = E[I(X)] = E[-\ln(P(X))]; \quad (16)$$

čia E - atsitiktinio dydžio pasirodymo tikimybė, I žymi informacijos duomenų srautą, kuriame pasirodo atsitiktinis dydis X .

$$H(X) = \sum_i P(x_i)I(x_i) = \quad (17)$$

$$= - \sum_i P(x_i) \log_b P(x_i) = - \sum_i \frac{n_i}{N} \log_b \frac{n_i}{N} = \log_b N - \frac{1}{N} \sum_i n_i \log_b n_i ;$$

$$\text{Jei } p(x_i) = 0, \text{ tai } \lim_{p \rightarrow 0^+} p \log(p) = 0;$$

čia b -naudojamo logaritmo pagrindas (dažniausiai tai 2 (dvejetainis logaritmas), Eulerio skaičius $e = 2,71828 \dots$ (natūrinis logaritmas) arba 10 (dešimtainis logaritmas)).

Entropijos H matavimo vienetai gali būti *bitai* ($b = 2$), *natai* ($b = e$) (entropijos *natūrinio* logaritmo matavimo vienetas, lygus ~ 1.44 bito ir ~ 0.434 *bano*) bei *banai* (pavadinimas kilo iš angliško žodžio *deciban* (dešimtainis skaičius, dešimtainio logaritmo 1/10 dalis) atmetus *SI* matavimo sistemoje naudojamą priešdėlį *Deci-*, nors šie vienetai ir nepriklauso *SI* matavimo sistemai).

Kaip matome, skaičiavimai pagal (17) formulę yra sudėtingi. Entropija informacijos teorijoje- tai dydis, naudojamas apibrėžti atsitiktinių duomenų reikšmių pasiskirstymą jų sraute [14], todėl šiame kontekste paprastai entropija išreiškiama racionalizuota Šenono teoremos [15] formule, kuri kiekybiškai įvertina tikėtiną reikšmės pasirodymą duomenų sraute ir yra apskaičiuojama pagal (18) formulę:

$$H = - \sum_i p_i \log_b p_i \quad (18)$$

čia p_i – simbolio reikšmės i pasirodymo duotame (analizuojamame) simbolių sraute, tikimybė.

Paanalizuokime literatūroje [15] minimą paprastą pavyzdį- skaitmeninį signalą, kurio duomenų seka susideda iš dviejų bitų (X, Y) įvesties ir dviejų bitų (X ir Y , X ar Y) išvesties. Darant prielaidą, kad du įvesties bitai X ir Y turi tarpusavyje nepriklausomas 50% pasirodymo tikimybes, tuomet kiekviena iš (0,0), (0,1), (1,0), ir (1,1) kombinacijų, turi pasirodymo tikimybę, lygią 25%. Informacijos teorijoje procesas, sukiantis atsitiktinius įvykius, vadinamas *šaltiniu* (įvestis). Šenono entropija tokiam *šaltiniui* užrašoma taip (19):

$$H(X, Y) = 4 \left(-\frac{1}{4} \log_2 \frac{1}{4} \right) = 2 \quad (19)$$

Gautasis rezultatas parodo koku bitų skaičiumi galima užkoduoti vieną kombinaciją. Kadangi šiuo atveju visų kombinacijų pasirodymo tikimybė yra vienoda, vienai kombinacijai užkoduoti vidutiniškai bus sunaudoti du bitai, o tai reiškia, kad kodavimas bus neefektyvus. Tačiau priklausomai nuo įvykių eiliškumo, jų tarpusavio priklausomybės ir kitų veiksnių, dažnai įmanoma rasti tokį kodą, kuris įvykių seką (statistiniu atveju) išreikštų mažesniu bitų skaičiumi. Analizuojamos sekos atveju, Šenono entropija galimoms išėjimo kombinacijoms (0,0), (0,1) ir (1,1), su atitinkamai 25%, 50% ir 25% pasirodymo tikimybe, apskaičiuojama pagal (20) formulę:

$$H(XUY, X \cap Y) = 2 \left(-\frac{1}{4} \log_2 \frac{1}{4} \right) - \frac{1}{2} \log_2 \frac{1}{2} = 1 + \frac{1}{2} = 1 \frac{1}{2} \quad (20)$$

Apskaičiavus rezultatą pagal (19) formulę, matome, jog dabar vienai dviejų bitų kombinacijai užrašyti vidutiniškai pakaks 1,5 bito. Nesunku paskaičiuoti, kiek bitų užims tokia koduota seka, jei reikia užkoduoti tris dviejų bitų kombinacijas: $L_{bitų} = 3 \times 1.5 = 4.5$ bitų. Čia L yra koduotos sekos ilgis bitais.

Iš gautų rezultatų matome, jog žinant tam tikrų simbolių sekų pasirodymo tikimybę šaltinyje, galime jo teikiamą informaciją suglaudinti $\frac{6}{4.5} \approx 33\%$ ir taip sutaupyti duomenų laikmenos ar perdavimo kanalo pralaidumo. Kuo dažniau sekoje pasitaiko tam tikros kombinacijos, tuo mažiau bitų sunaudojama koduojamai informacijai užrašyti, taigi ir glaudinimo efektyvumas didesnis. Šis glaudinimo kriterijus ypač svarbus, apdorojant archyvuojamų ar didelių apimčių siauros paskirties duomenų masyvus. Duomenų srauto entropijos analize taip pat paremti Hafmano ir aritmetinis kodavimas.

1.8.2. Algoritmų sudėtingumas

Kaip teigiama šaltinyje [18], paprasčiausi deterministiniai algoritmai, turi apibrėžtą sudėtingumą visais atvejais. Grupavimo metodų sudėtingumas dažniausiai yra $O(n)$. Sudėtingesnius metodus realizuojantys algoritmai dažnai reikalauja daug resursų atminties ir laiko atžvilgiu, todėl dažnai tokie algoritmai pritaikomi dirbti su fiksuoto ilgio posekais (su apribojimu euristikos apimčiai). Tokiu atveju algoritmo sudėtingumą galima laikyti $O(n)$, nes skaičiavimų sąnaudos beveik tiesiškai priklauso nuo glaudinamų duomenų apimties. Tokie apribojimai euristikos apimčiai

tam tikrais atvejais turi lemiamos įtakos koduotų duomenų entropijai (pavyzdžiui fraktalinio arba praradiminio glaudinimo atveju). Todėl šis (sudėtingumo) kriterijus aktualesnis analizuojant metodų teorines galimybes.

Dažnai, realizuojant algoritmą, tenka jį įvertinti numatant apdorojamų posekių ilgus, išankstinių lentelių apimtis ir technines įrangos galimybes. Priklausomai nuo įrangos architektūros, kai kada efektyviau duomenis saugoti glaudžiuoju kodu (pavyzdžiui, jei jie laikomi mažos spartos laikmenoje). Šiuo atveju labai svarbu parinkti metodą ir jo parametrus, kad fragmento paieška, kodavimas ir dekodavimas nesueikvotų daugiau sistemos išteklių nei sutaupoma juos koduojant.

1.8.3. Algoritmų tinkamumas

Kai kurie metodai (LZW, aritmetiniai, Hafmano) suglaudintų duomenų meta informaciją įrašo duomenų sekos pradžioje [18]. Kitų metodų, tokių, kaip JPEG, JPEG2000 (ar kiti diskrečių bangelių transformacijos metodai), archyvavimo, šifravimo algoritmų suglaudintų sekų n -ojo fragmento dekodavimui pakanka turėti dekoduatą $(n - k)$ fragmentą. Pastaroji kodo savybė gali padidinti algoritmo efektyvumą, tačiau tai riboja to metodo taikymą konkrečiose srityse.

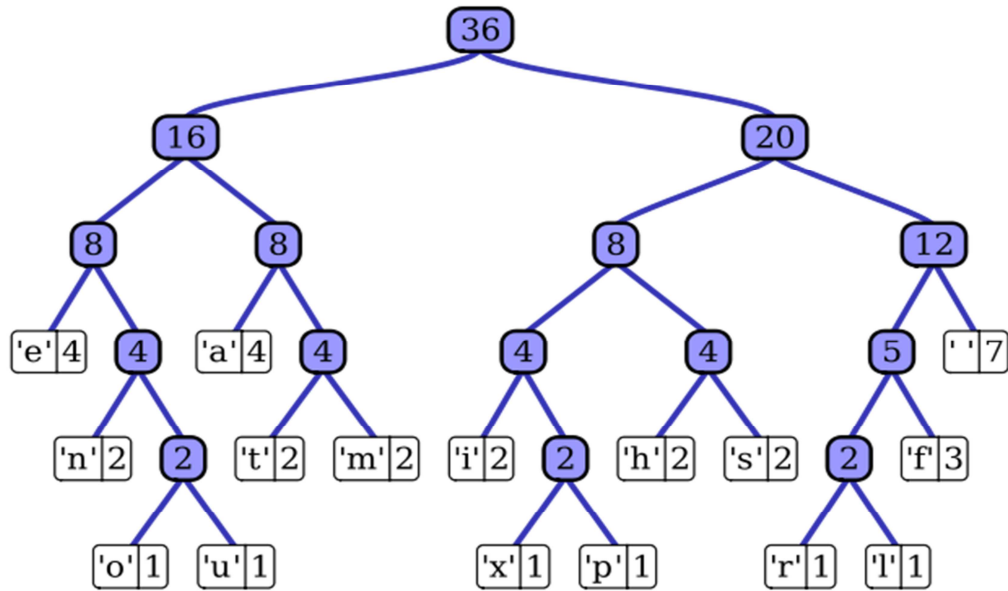
Nesrautiniai kodavimo algoritmai sinchronizavimo metaduomenis įterpia į suglaudintus duomenis, tačiau tai mažina metodo efektyvumą, duomenų struktūra tampa sudėtingesnė. Taigi prieš realizuojant tam tikro tipo algoritmą, reikia atsižvelgti į numatomą jo panaudojimo sritį ir jos specifiką, bei svarbiausius kriterijus šioje srityje (pvz.: algoritmo veikimo greitis realaus laiko sistemose, ar suspaudimo lygis duomenų archyvavimo srityje). Kadangi entropinio duomenų glaudinimo efektyvumas priklauso nuo šaltinio duomenų ir jų entropijos, taikant jį būtina atsižvelgti į kontekstą, kuriame jis bus naudojamas. Darbe nagrinėjamu atveju, vietoje JPEG standarte naudojamo RLE, yra taikomas A&A entropinis algoritmas. JPEG yra specifinis dėl jo sekose ypač dažno skaičiaus „0“ ir jo sudaromų sekų (žr. 1.7.2.4 posk.). Pastaroji savybė jau panaudota adaptuojant RLE. Siekiant, kad A&A algoritmo taikymas būtų tikslingas, jis turi išnaudoti dar neišnaudotas savybes ir duoti didesnę efektyvumą.

1.9. Hafmano kodavimo algoritmas

Aritmetiniai kodavimo algoritmai vienam šaltinio sekos simboliui priskiria trumpesnę jį atitinkantį kodą, atsižvelgiant į to simbolio pasikartojimų dažnį. Taip sunaudojama mažiau bitų vienam simboliui koduoti ir koduojami tik sekoje esantys simboliai. Tokio tipo algoritmai yra adaptyvūs, jų simboliui priskiriami kodai kiekvienu atveju yra unikalūs ir neturi fiksuoto dyžio bitais [16]. Panašus, tačiau kitoks yra Hafmano (*angl. David Albert Huffman (1925– 1999)*) informacijos teorijoje ir kompiuterijoje laikomas entropiniu žodyniniu kodavimu. Jis yra naudojamas duomenų suglaudimui be informacijos praradimo [17]. Skirtumas nuo aritmetinių yra tas, kad Hafmano kodavimas grindžiamas minimalaus šaltinio simbolių sekos medžio sudarymu. Šaltinio sekos simboliai išrikiuojami dažnėjimo tvarka, taip randamos mažiausiai pasikartojančios reikšmės ir suformuojamas medis [2] [17]:

- Retesnis simbolis yra pirmas lapas;
- Dažnesnis yra antras lapas;
- Šaknis yra abiejų lapų dažnių suma:
 - Reikšmės pašalinamos iš sekos, šaknis patalpinama į surikiuotą seką.
 - Procesas kartojamas, kol lieka tik vienas elementas, jis yra Hafmano medžio šaknis.

Tarkime naudojant Hafmano metodą, reikia užkoduoti anglišką sakinį „this is an example of a Huffman tree“ (*liet. „tai Hafmano medžio pavyzdys“*). Tuomet sudaromas Hafmano medis (1.20 pav.):



1.20 pav. Hafmano medis (www.wikipedia.org)

Sudaryto medžio duomenys surašomi į simbolių lentelę (1.21 pav.), kurioje įrašomas simbolio pasikartojimų skaičius ir jam priskirtas kodas.

Simolis	Pasikartojimų dažnis	Kodas
Tarpas	7	111
e	4	000
a	4	010
f	3	1101
n	2	0010
t	2	0110
m	2	0111
i	2	1000
h	2	1010
s	2	1011
o	1	00110
u	1	00111
x	1	10010
p	1	10011
r	1	11000
l	1	11001

1.21 pav. Hafmano medžio duomenų lentelė

Tokio sakinio užkodavimas, reikalauja 135 bitų, vietoje nekoduotam sakiniui reikalingų 288 bitų (36 simboliai po 8 bitus). Čia neįskaičiuota Hafmano medžio struktūrai saugoti reikalinga atminties dalis, laikant, kad struktūra dekoduojant yra žinoma.

Formaliame Hafmano metodo apibrėžime naudojami dydžiai šaltinio (įvesties) simbolių sekai apibūdinti:

- Abėcėlė $A = \{a_1, a_2, \dots, a_n\}$, tai simbolių abėcėlė, kurios dydis η .
- Rinkinys $W = \{w_1, w_2, \dots, w_n\}$, tai (teigiamas) visų simbolių svoris, proporcingas tų simbolių pasirodymo dažniui.

Apibrėžime naudojami dydžiai išvesties (koduotai) simbolių sekai apibūdinti:

- Kodas $C(A, W) = \{c_1, c_2, \dots, c_n\}$, tai dvejetainių kodavimo žodžių rinkinys, kuriame c_i yra kodo žodžių kiekis intervale $1 \leq i \leq n$.

Tikimybė, jog koduotos šaltinio sekos C kelio, einančio per sudarytojo Hafmano medžio šaknines viršūnes, ilgis bus L , apskaičiuojama pagal (21) formulę:

$$L(C) = \sum_{i=1}^n w_i \times \text{ilgis}(c_i), \text{ kai } L(C) \leq L(T) \text{ kiekvienam kodui } T(A, W) \quad (21)$$

Dekoduojant, žingsniai vykdomi atvirkštine tvarka- Hafmano metodu sudaryto medžio kairįjį lapą identifikuojant kaip „0“, dešinįjį- kaip „1“. Taigi bet kokį simbolių galima pasiekti, naudojant Hafmano kodą, kuris yra trumpesnis už, sakykim 8 bitų ASCII kodą. Savaime suprantama, kad kodo iššifravimui būtina pati Hafmano medžio struktūra [2] (1.21 pav.).

1.10. LZW kodavimo algoritmas

Kaip ir Hafmano, LZW (*angl. Lempel–Ziv–Welch*), yra entropinis žodyninis kodavimas, glaudinimui naudojantis kodavimo pertekliškumą [19]. LZW yra JAV mokslininko Teri Velšo (*angl. Terry Welch*) patobulintas LZ78 (publikuoto 1978 m.) algoritmo variantas. LZ78 algoritmą sukūrė du Izraelio mokslininkai- Abrahamas Lempelis (*Abraham Lempel (1936)*) ir Džekobas Zivas (*Jacob Ziv (1931)*) [19].

Sutrumpinus šaltinyje [20] pateiktą informaciją, apie LZW suspaudimo algoritmą, jis gali būti apibūdintas kaip nuosekliai nuskaitantis duomenis algoritmas, kuris sukuria kodų vertimo lentelę (žodyną), sudarytą iš nuorodų (indeksų). Kiekvienas naujas indeksas yra įtraukiamas į paminėtą kodų lentelę, o to indekso reikšmė įrašoma į išvedimo srautą (seką). Jeigu, toliau nuskaitant duomenis, sutinkamas indeksas, kuris jau yra kodų lentelėje, jo reikšmė įtraukiama į išvedimo srautą, o kodų lentelėje formuojamas naujas indeksas, apimantis daugiau simbolių. Atliekant šį procesą, pasiekiamas duomenų suspaudimo efektas.

Abstrakčiai LZW algoritmą galima į tokius žingsnius:

1. Inicijuoti kodų vertimo lentelės (žodyno) sudarymą, kurioje kiekviena seka yra simbolio ilgio;
2. Surasti ilgiausią simbolių seką w , kuri atitinka analizuojamą (įvesties) seką;
3. w sekos indeksą (kodą) įrašyti į išvesties seką ir pašalinti w seką iš įvesties sekos;
4. Į kodų vertimo lentelę įrašyti w seką su sekančiu simboliu ($w + 1$);
5. Vykdyti 2 žingsnį.

Kaip pavyzdį, panagrinėkime trumpą, šaltinyje [20] pateikiamą simbolių seką „TOBEORNOTTOBEORTOBEORNOT#“. Ši seka specialiai sumodeliuota LZW algoritmo efektyvumui parodyti. Paprastai pasikartojimai tekste nėra labai tikėtini, todėl būtų reikalingas labai ilgas tekstas, kad algoritmas veiktų efektyviai. Pateiktame pavyzdyje analizuojama tik paprastojo teksto (*angl. plain text*) simbolių- didžiųjų raidžių seka. Simbolis „#“ naudojamas kaip eilutės pabaigos žymeklis. Taigi, jei naudojama lotyniška tik didžiųjų raidžių abėcėlė, ją sudarančių 26-ių simbolių ir pabaigos žymeklio „#“ užkodavimui reikės ($2^5 = 32 > 27$) t.y. penkių bitų. Žymeklį „#“ kodavime patariama koduoti pirmąjį, priskiriant „0“ kodą. Taip galima išvengti klaidos, kuri gali atsirasti, jei pvz.: penkių bitų nebepakaks koduotos sekos simboliams užrašyti. Tuomet algoritmas pradės naudoti šešis bitus simboliams koduoti ir taip žymekliui priskirtas kodas gali būti perimtas kito simbolio užkodavimui. Be eilutės pabaigos žymeklio, tam tikrais atvejais, algoritmo ciklas tęsis „amžinai“.

Nagrinėjamą seką, kompiuteris apdoros kaip bitų eilutę. Pradedant vykdyti LZW algoritmą, inicijuojama lentelė (1.22 pav.) tik su sekoje esančiais simboliais. Kiekvienam simboliui priskirtas jį atitinkantis kodas.

Simbolis	Dvejetainis kodas	Dešimtainis kodas
#	00000	0
A	00001	1
B	00010	2
C	00011	3
D	00100	4
E	00101	5
F	00110	6
G	00111	7
H	01000	8
I	01001	9
J	01010	10
K	01011	11
L	01100	12
M	01101	13
N	01110	14
O	01111	15
P	10000	16
Q	10001	17
R	10010	18
S	10011	19
T	10100	20
U	10101	21
V	10110	22
W	10111	23
X	11000	24
Y	11001	25
Z	11010	26

1.22 pav. Pradinė (jėjimo sekos) kodų vertimo lentelė

Kodavimo procesas (1.23 pav.) vykdomas jau aprašytais žingsniais: buferio įvesties simbolių seka yra w tol, kol seka $w +$ kitas simbolis ($w + 1$) nėra žodyne. Jei ($w + 1$) nėra žodyne, tada išmesti w sekos kodą, ir pridėti $w +$ sekantis simbolis ($w + 1$) į žodyną. Pradėti vykdymą iš naujo su kitu šaltinio simboliu.

Dabartinė seka	Sekantis simbolis	Išvestis		Išplėstinis žodynas	Pastabos
		Dešimtainis kodas	Dvejetainis kodas		
NULL	T				
T	O	20	10100	27: TO	27-tai pirmasis laisvas dešimtainis kodas
O	B	15	01111	28: OB	
B	E	2	00010	29: BE	
E	O	5	00101	30: EO	
O	R	15	01111	31: OR	
R	N	18	10010	32: RN	32 elemento užkodavimui reikia 6 bitų, sekančiam išėjimo simboliui naudoti 6 bitus.
N	O	14	001110	33: NO	
O	T	15	001111	34: OT	
T	T	20	010100	35: TT	
TO	B	27	011011	36: TOB	
BE	O	29	011101	37: BEO	
OR	T	31	011111	38: ORT	
TOB	E	36	100100	39: TOBE	
EO	R	30	011110	40: EOR	
RN	O	32	100000	41: RNO	
OT	#	34	100010		# algoritmas sustoja; įrašoma dabartinė seka
		0	000000		eilutės pabaigos žymeklis

1.23 pav. Pavyzdinė LZW algoritmo kodavimo schema

Atkuriant duomenis (1.24 pav.), nuskaitomas suspaustas masyvas bei kuriama kodų vertimo lentelė. Pilna vertimo lentelė sudaroma, nuskaičius visą suspaustą masyvą. Pradiniai duomenys atkuriami, suspaustame sraute nuorodas pakeičiant atitinkamomis reikšmėmis.

Išvestis		Išvesties seka	Nauja indekso reikšmė žodyne				Pastabos
Dešimtainis kodas	Dvejetainis kodas		Pilna		Spėjimas		
10100	20	T			27:	T?	
01111	15	O	27:	TO	28:	O?	
00010	2	B	28:	OB	29:	B?	
00101	5	E	29:	BE	30:	E?	
01111	15	O	30:	EO	31:	O?	
10010	18	R	31:	OR	32:	R?	Paskutinis simbolis, užkodotas 5-iais bitais
001110	14	N	32:	RN	33:	N?	nuskaitoma po 6 bitus/ žodžius
001111	15	O	33:	NO	34:	O?	
010100	20	T	34:	OT	35:	T?	
011011	27	TO	35:	TT	36:	TO?	
011101	29	BE	36:	TOB	37:	BE?	elementas iš „TO“ ir 1-ojo simbolio (B) iš
011111	31	OR	37:	BEO	38:	OR?	sekančios gautos koduotos sekos
100100	36	TOB	38:	ORT	39:	TOB?	
011110	30	EO	39:	TOBE	40:	EO?	
100000	32	RN	40:	EOR	41:	RN?	
100010	34	OT	41:	RNO	42:	OT?	
000000	0	#					eilutės pabaigos žymeklis

1.24 pav. Pavyzdinė LZW algoritmo dekodavimo schema

Pateikto pavyzdžio nekoduotos duomenų sekos ilgis= 25 simboliai \times 5 bitai/ simboliui = 125 bitai, koduotos sekos ilgis= (6 kodai \times 5 bitai/kodui) + (11 kodų \times

6 bitai/kodui) = 96 bitai. Naudojant 1.6.1 skyriuje nurodytą (5) formulę galima įvertinti glaudinimo lygį:

$$\alpha = \frac{I}{I'} = \frac{125}{96} \approx 1,302$$

Koeficientas α parodo, jog koduoti duomenys, šiuo atveju, užima maždaug 1,3 karto mažiau vietos, nei neglaudinti duomenys.

Jei tokių tekstinių duomenų seka būtų ilgesnė, sudarytojo žodyno kodai apimtų ilgesnes teksto sekcijas, kas leistų pasikartojimus koduoti dar efektyviau.

1.11. Run-Length Encoding algoritmas

Sekos ilgio kodavimas (angl. RLE) yra duomenų suspaudimo algoritmas, palaikomas daugumos bitų žemėlapių (angl. *Bitmap*) failų formatų, tokių kaip TIFF (angl. *Tagged Image File Format*), BMP (angl. *BitMap Picture*) ir PCX (angl. *Personal Computer Exchange*). RLE naudojamas bet kokio duomenų tipo glaudinimui, nepriklausomai nuo informacijos turinio, tačiau informacijos turinio duomenys turi didelę įtaką RLE efektyvumui [16]. Nors palyginus su sudėtingesniais entropinio kodavimo algoritmais, RLE negali pasiekti didelio duomenų glaudinimo efektyvumo, tačiau jis paprastai realizuojamas ir greitai vykdomas. Tai gera alternatyva sudėtingiems glaudinimo algoritmams ir dažniausiai efektyviau naudoti šį algoritmą nei palikti duomenų sekas visiškai nekoduotas.

RLE veikimo principas- sumažinti fizinį pasikartojančių simbolių sekos dydį. Šie pasikartojantys simboliai, kitaip seka, koduojama dviem baitais (angl. *Byte*). Pirmasis baitas nusako pasikartojančių simbolių sekos ilgį ir yra vadinamas *sekos skaitliuku*. Praktikoje, užkoduotoje sekoje gali būti nuo 1 iki 128 ar 256 simbolių. Sekos skaitliuko reikšmė dažniausiai žymi simbolių skaičių minus vienetas (reikšmė ribose tarp 0 ir 127 ar 255). Antrasis baitas yra skaičiaus, iš kurių sudaryta seka, reikšmė (ribose nuo 0 iki 255). Jis vadinamas *sekos reikšme*. Pavyzdyje (1.25 pav.)1.25 pav. nekoduota 15 simbolių „A“ seka, būtų 15-os baitų apimties:

```
AAAAAAAAAAAAAAAA
```

1.25 pav. Nekoduota „A“ simbolių seka

Jeigu būtų panaudotas RLE kodavimas, tokios sekos apimtis sumažėtų iki 2-jų baitų (1.26 pav.):

```
15A
```

1.26 pav. RLE koduota „A“ simbolių seka

Kodas, gautas panaudojus RLE, žymintis užkoduotą 15 simbolių „A“ seką, vadinamas RLE paketu. Čia pirmasis baitas „15“ yra sekos ilgio skaitliukas, žymintis simbolių pasikartojimų skaičių. Sekantis baitas, simbolis „A“ yra pasikartojanti sekos reikšmė.

RLE paketas, sugeneruojamas kiekvieną kartą, kai pasikeičia sekos simbolis arba tuomet, kai sekos ilgio skaitliukas pasiekia maksimalią įmanomą reikšmę. Priimkime, kad 15 simbolių seka (1.27 pav.), dabar sudaryta iš kitų simbolių:

```
AAAAABBBCCCCD
```

1.27 pav. Nekoduota „A, B, C ir D“ simbolių seka

Kaip matosi iš (1.28 pav.) paveikslėlio, panaudojus RLE kodavimą, seką galima užkoduoti keturiais 2 baitų paketais:

6A3B5C1D

1.28 pav. RLE koduota „A, B, C ir D“ simbolių seka

Po sekos ilgio kodavimo (RLE), 15 baitų sekos duomenų apimtis užimtų tik aštuonis baitus, neprarandant pirminių duomenų t.y. su galimybe pilnai atstatyti originalią duomenų seką. Šiuo konkrečiu atveju, RLE panaudojimas beveik dvigubai sumažino duomenų kiekį.

Tam tikruose duomenų tipuose ilgos sekos pasitaiko retai. Pavyzdžiui ASCII paprastojo (*angl. plain*) teksto koduotė retai sudaro ilgas simbolių sekas. Prieš tai pateiktame pavyzdyje (1.28 pav.), sekos paskutinis simbolis buvo „D“, ir toks simbolis sekoje buvo vienintelis, tačiau RLE algoritmo realizacijoje, vieno simbolio seka jau yra seka. Tam, kad būtų naudinga naudoti RLE kodavimą, tiek sekos ilgio skaitliuko reikšmė tiek ir pasikartojanti sekos reikšmė, turi būti užrašytos mažiausiai dviejų simbolių sekai. RLE kodavimo metu, reikia bent jau dviejų pasikartojančių simbolių sekos, kad nebūtų užimama papildomai vietos t.y. du baitai užkoduojami tais pačiais dviem baitais, tiesiog sudedant juos į RLE paketą. Iš pateiktos informacijos, seka išvada, jog vieno simbolio (vieno baito) seka, koduojant RLE, užima daugiau vietos, nei pats simbolis, nes kodavimui panaudojamas dviejų baitų RLE paketas, vietoje vieno baito, reikalingo užkoduoti patį simbolį. Dėl tos pačios priežasties, dviejų simbolių seka po RLE kodavimo užima tiek pat vietos, kiek ir nekoduojant.

Paveikslėlyje (1.28 pav.) pateiktoje simbolių sekoje, paskutiniam simboliui „D“ užkoduoti, RLE algoritmas sunaudojo du baitus, vietoje vieno. Tačiau šis nuostolis mažai įtakojo bendrą suspaudimo efektyvumą, nes sekoje buvo ilgos kitų simbolių sekos. Tam, kad išsiaiškinti, kaip RLE gali padidinti (padvigubinti) duomenų apimtį, panagrinėkime dar vieną simbolių seką, pateiktą 1.29 pav. :

ABCDEFGHIYJKLM

1.29 pav. Seka prieš RLE kodavimą

Naudojant RLE kodavimą, ši seka būtų užkoduota (1.30 pav.) kaip:

1A1B1C1D1E1F1G1H1I1Y1J1K1L1M

1.30 pav. Seka po neefektyvaus RLE kodavimo

Iš pateikto pavyzdžio nesunku apskaičiuoti, jog iš keturiolikos baitų apimties sekos, panaudojus RLE, gaunama užkoduota seka, užimanti dvigubai daugiau vietos.

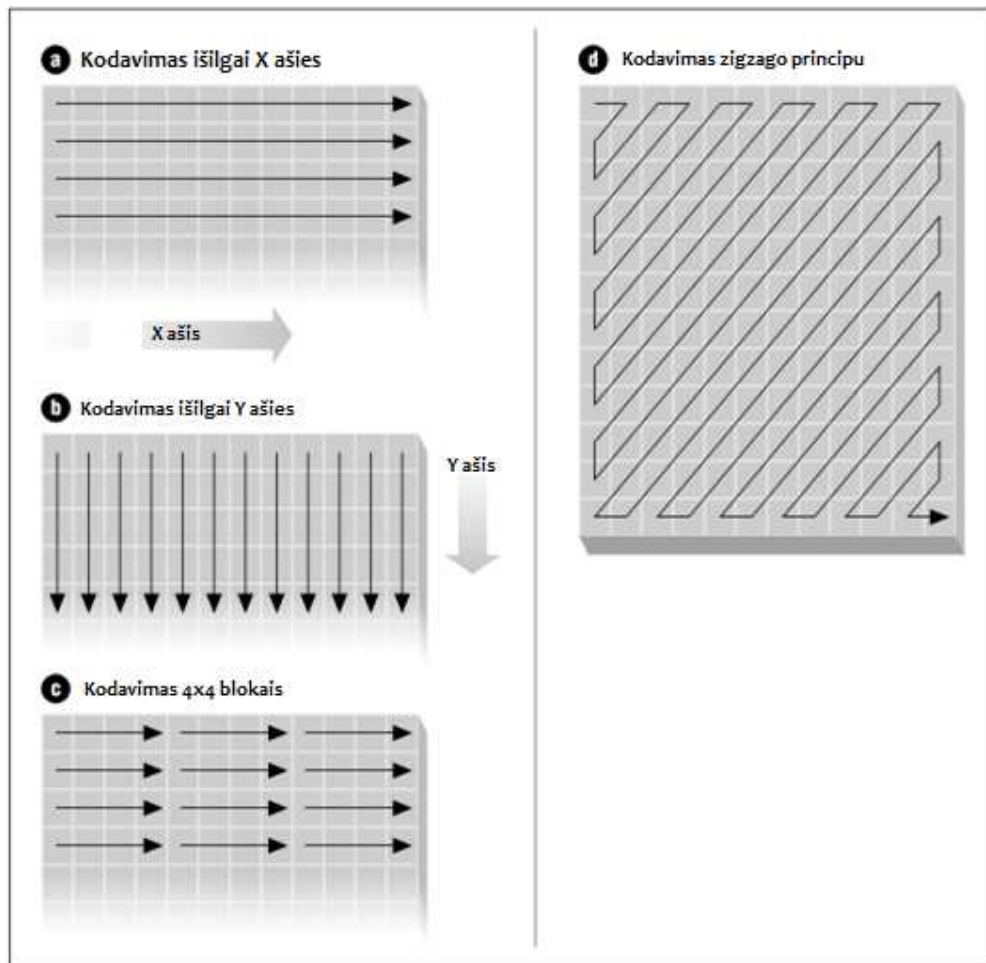
RLE veikimo principai yra paprasti ir greiti, tačiau šio algoritmo suspaudimo efektyvumas labai priklauso nuo koduojamų duomenų tipo. Juodai balti paveikslėliai, kuriuose daugiausiai baltos spalvos, tokie kaip knygos puslapiai, naudojant RLE, bus koduojami labai gerai. Tokiuose paveikslėliuose yra daug vienarūšių duomenų, šiuo atveju tokios pat spalvos.

Kitokio tipo paveikslėliai, tokie kaip fotografijos, kuriuose daug ir įvairių spalvų, kontrastų, RLE kodavimas nebus itin efektyvus. Tai lemia paveikslėlio sudėtingumas dėl spalvų įvairovės. Tokia įvairovė glaudžiai susijusi su ilgų (šiuo atveju tos pačios spalvos) sekų nebuvimu.

1.11.1. Sekos ilgio kodavimo variantai

Galimi keli sekos ilgio kodavimo variantai. Paveikslėlių duomenų sekų ilgis paprastai koduojamas nuosekliu procesu, kuris dviejų dimensijų duomenų žemėlapiu (paveikslėlio) duomenis apdoroja kaip vienmatį srautą. Nuoseklaus proceso metu, bitų žemėlapis (*angl. BitMap*) apdorojamas (koduojamas) pradedant nuo viršutinio kairiojo kampo ir koduojant nuo kairės į dešinę išilgai kiekvienos nuskaitomos eilutės (ašis X) ir taip tęsiant link apatinio dešiniojo bitų žemėlapiu kampo (1.31 pav., a). Alternatyviais atvejais, RLE veikimo schema gali būti sudaryta duomenų nuskaitymui nuo viršaus į apačią, per visą bitų žemėlapiu aukštį (Y ašis), išilgai bitų stulpelių (1.31 pav., b). Taip

pat galima naudoti schemą kurioje sekos nuskaitymas vykdomas dviejų matavimų bitų (pikselių) blokais (1.31 pav., c). Atsižvelgiant į tai, jog RLE nagrinėjamas kaip JPEG standarto dalis, aktuali RLE sekos nuskaitymo schema yra pagal bitų žemėlapiu (JPEG atveju 8x8 bloko, žr. 3 Skaidymas į blokus) įstrižainę, vadinamuoju zigzago principu (žr. 6 Zigzaginis nuskaitymas). Pastaroji schema pavaizduota paveikslėlio 1.31 pav., d dalyje.



1.31 pav. RLE sekų sudarymo variantai

Dar vienas, rečiau sutinkamas RLE variantas, yra sekos ilgio kodavimas su duomenų praradimu. Standartiškai RLE algoritmai yra algoritmai be informacijos praradimo. Tačiau atsisakant duomenų kodavimo procese (dažniausiai prilyginant nuliui vieną ar du mažiausiai svarbius bitus kiekviename pikselyje), galima žymiai padidinti suspaudimo efektyvumą, be žymios neigiamos įtakos spalviškai įvairaus paveikslėlio vaizdai. Ši RLE schema, veikia efektyviai tik realaus pasaulio paveikslėliuose (fotografijose), kuriuose yra daug subtilius pokyčius atspindinčių pikselių (jų reikšmių).

1.11.2. Sekos ilgio kodavimo specifika

RLE algoritmo efektyvumas labai priklauso nuo pačių duomenų (jų struktūros). Kuo duomenų entropija (žr. 1.8 skyrių) bus mažesnė, tuo kodavimas bus efektyvesnis. Maža duomenų entropija užtikrina, kad koduojamose duomenų eilutėse bus RLE kodavimui reikalingos pasikartojančių simbolių sekos. Taigi prieš taikant RLE algoritmą, reikia išanalizuoti įėjimo duomenų tipą, panaudojimo sritį, išsiaiškinti ar statistiškai tos srities generuojamose duomenų sekose yra pakankamai daug pasikartojančių simbolių.

Kodavimui naudojant RLE algoritmą, jį reikia realizuoti taip, kad jis sustotų ties kiekvienos nuskaitymos paveikslėlio eilutės, stulpelio, 4x4 bloko ar zigzago principu sudarytos eilutės

(atitinkamai 1.31 pav. a, b, c ir d dalys) pabaiga ir ją pažymėtų. Toks algoritmo panaudojimas turi kelis privalumus. Vienas iš jų yra tas, jog vienu metu koduojant tik vieną eilutę, reikalingas minimalius buferio dydis. Toks panaudojimas taip pat apsaugo nuo problemos, pasitaikančios naudojant kryžminį kodavimą.

1.11.3. Kryžminis kodavimas

Kryžminio kodavimo neigiamas efektas pasireiškia tuomet, kai kodavimo procese „pametamas“ skirtumas tarp originalių eilučių. Tai atsitinka dėl to, kad eilučių nuskaitymas ir apdorojamas vykdomas kryžminiu būdu. Dažniausiai tai lemia sekų išėjimas už eilutės ribų. Jei atskirų eilučių duomenys RLE kodavimo metu yra apjungiami (vienos eilutėje susidariusi seka, tęsiasi ir kitoje (-se) eilutėse), žymės, kur viena eilutė baigėsi ir prasidėjo kita, yra neįmanoma surasti arba jos suradimas užima labai daug laiko.

Kryžminis kodavimas vis dėl to naudojamas, tačiau jis turi daugiau trūkumų nei privalumų. Jo panaudojimas gali sutaupyti keletą papildomų duomenų apimties baitų, tačiau komplikuojamas dekodavimo procesas, išauga laiko sąnaudos. Kadangi bitų žemėlapių failų formatuose duomenų glaudinimui naudojami tik entropiniai algoritmai, tai tokios technikos nauda, dėl jos didesnio suspaudimo lygio, yra didesnė nei gaunama naudojant nuoseklaus nuskaitymo- kodavimo techniką. Nors nemažai failų formatų specifikacijose tiksliai nurodyta, jog kodavimas RLE algoritmu turi būti atliekamas nuskaitymą ir koduojant kiekvieną eilutę atskirai, didelėje dalyje programinės įrangos kodavimas procesas vykdomas kryžminiu būdu, sudarant ir koduojant informaciją, kaip vientisą duomenų srautą ir nepaisant nuskaitymų eilučių ribų. Tokiais atvejais iškyla suderinamumo problema. Tarkim RLE užkoduotas paveikslėlis, gali būti atidaromas naudojant vieno gamintojo programinę įrangą (kurioje naudojamas kryžminis (de)kodavimas). Jei kita programinė įranga kodavimui/ dekodavimui naudoja nuoseklų nuskaitymą (eilutės turi ribas), ji negalės tinkamai dekoduoti tokio paveikslėlio. Norint išvengti nesuderinamumo problemos, programinėje įrangoje patariama naudoti kryžminį dekodavimą. Algoritmą reiktų realizuoti taip, kad koduota seka visada sustotų nuskaitytos eilutės pabaigoje ir tos eilutės pabaigą atitinkamai pažymėtų.

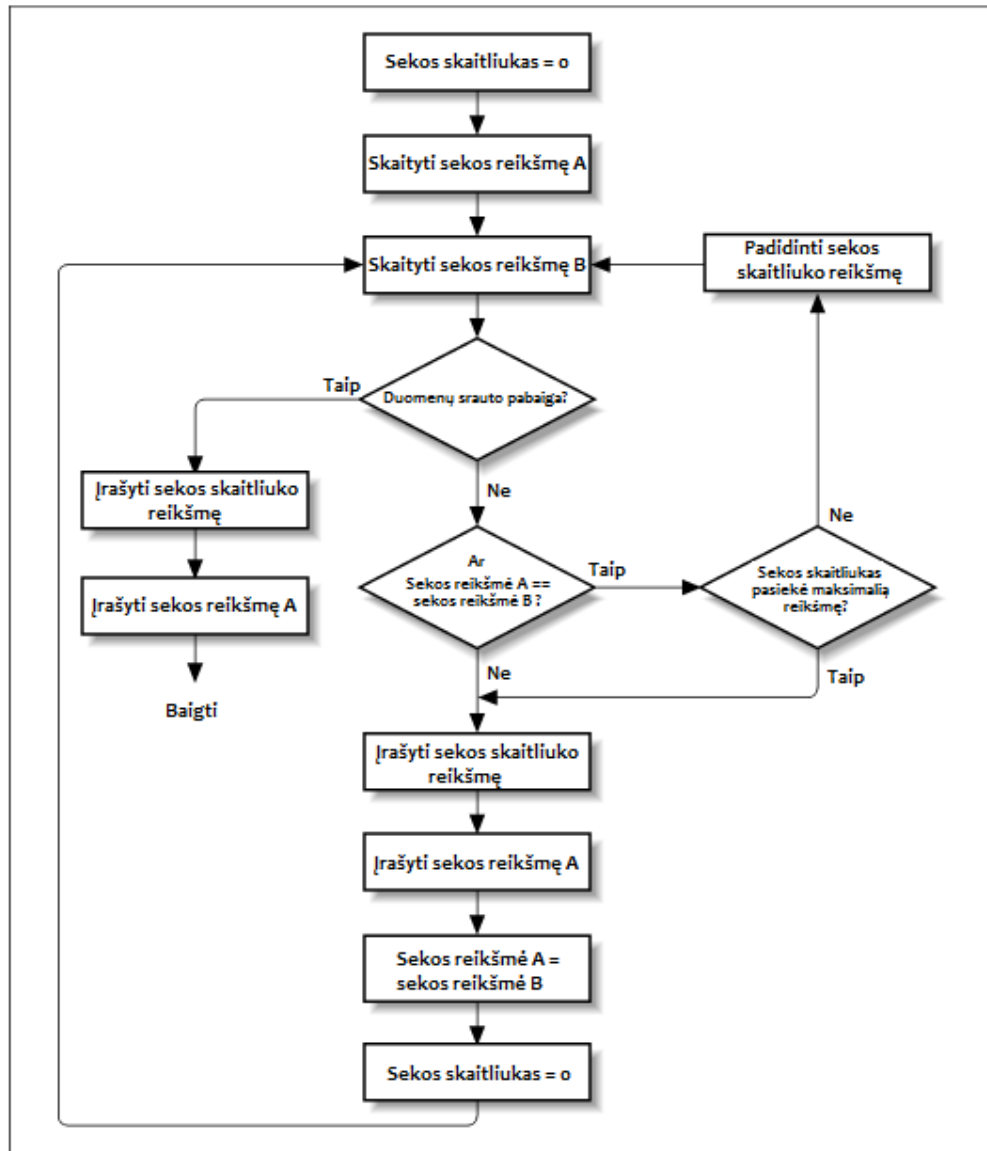
Kuomet RLE kodavimo algoritmas koduoja paveikslėlį, nuskaitytos eilutės pabaigos vieta pažymima žymekliu tam, kad dekodavimo algoritmas galėtų interpretuoti, kada buvo pasiekta tos eilutės pabaiga. Standartiškai, šis žymeklis tai unikalus paketas, apibrėžtas RLE specifikacijoje. Toks paketas negali būti supainiotas su jokių kitu RLE paketu. Eilutės (stulpelio) žymeklis paprastai yra tik vieno baito apimties, taigi bendros koduojamų duomenų apimties labai nepadidina [12].

Koduojant kiekvieną eilutę atskirai yra naudinga, kuomet aplikacijai užduota komanda atvaizduoti tik dalį paveikslėlio. Tarkime paveikslėlį sudaro 600 eilučių, bet atvaizduoti reikalinga tik eilutes nuo 290-osios iki 320-osios. Jeigu koduojant nebuvo įdėti eilučių pabaigos žymekliai, aplikacija prieš dekodavimą reikalingas 30 eilučių, turėtų dekoduoti eilutes nuo 1-osios iki 290-osios. Toks procesas yra neefektyvus, nes reikalauja papildomų laiko sąnaudų. Jei tokiu atveju buvo naudojamas kryžminis kodavimas, bet buvo sudėti eilučių žymekliai, tuomet aplikacijai beliktų nuskaityti koduotas eilutes, suskaičiuoti eilučių pabaigos žymeklius, kol bus pasiekta pirmoji reikalinga eilutė ir tuomet dekoduoti nuo jos.

Dar vienas galimas sprendimas norint greitai ir efektyviai surasti kryžminiu būdu užkoduotų eilučių pradžią, tai sudaryti koduotų eilučių lentelę. Tokia lentelė sudaroma iš elementų, apibrėžiančių kiekvieną atskirą originalią eilutę. Šis elementas nurodo poslinkį nuo nuskaitymo iki juo apibrėžiamos eilutės pradžios. Jei pavyzdžiui reikia surasti 25-ąją koduotą eilutę, dekodavimo algoritmui reikia tik nuskaityti poslinkio, saugomo 25-ajame lentelės elemente, reikšmę. Į lentelę taip pat galima surašyti ir baitų kiekį, kuris buvo sunaudotas užkoduoti kiekvieną eilutę. Šiuo atveju dekodavimo algoritmui reiktų apskaičiuoti poslinkį, sudedant lentelėje saugomas pirmųjų 24-ių elementų reikšmes. Analizuojamos 25-osios eilutės pradžia būtų sekantis baitas po apskaičiuotojo poslinkio.

1.11.4. Bitų, baito ir pikselio lygio RLE schemas

Bazinė RLE algoritmo sekų diagrama pavaizduota 1.32 pav. paveikslėlyje



1.32 pav. Bazinė RLE algoritmo sekų diagrama

Išskiriamos kelios RLE algoritmo panaudojimo sritys, kurios tarpusavyje skiriasi koduojamų duomenų tipu (pavyzdžiui skaitiniai ar raidiniai). RLE schemas, apibrėžiančios bitų žemėlapių tipo grafinių vaizdų kodavimą, yra skirstomos į klases pagal pagrindines savybes, kuriomis pasižymi koduojami duomenys. Išskiriamos trys pagrindinės duomenų klasės (lygiai), naudojamos grafikos vaizdų failų formatuose yra *bito*, *baito* ir *pikslio*. Jų kodavimui pasitelkiamas RLE. Apžvelkime kiekvieną iš klasių.

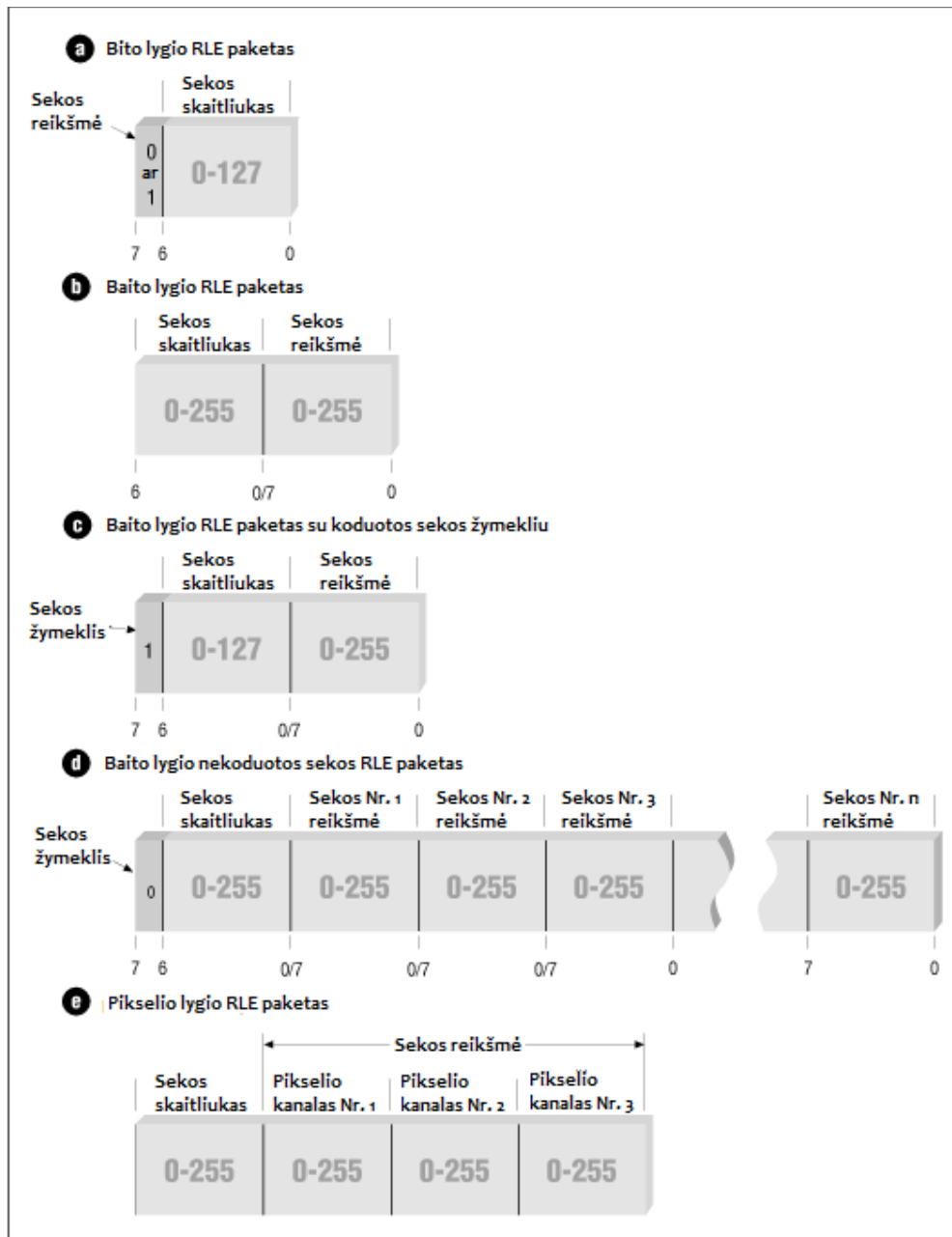
Bito lygio RLE schemeje kodavimo įranga sekas koduoja kaip gretimų bitų srautą, ignoruojant baitų ir žodžių ribas. Tokio tipo schemas efektyvu naudoti tik vienspalvių (juodai baltų), 1 bito paveikslėlių kodavimui, kadangi juose yra pakankamas vienodų pasikartojančių ir ilgas sekas sudarančių bitų kiekis. Pagal šią RLE schemą, sudaromi vieno baito paketai, kuriuose saugoma informacija apie užkoduotų bitų sekų ilgį (nuo 1 iki 128 bitų ilgio sekai). Paskutiniai mažiau svarbūs septyni paketo bitai nurodo sekos ilgio skaitliuko reikšmę minus vienetas. Svarbiausias bitas yra pirmasis, kuris nurodo sekos reikšmę t.y. 0 arba 1 (1.33 pav., a dalis). Ilgesnė, nei 128 pikselių seka išskaidoma į atskirus RLE užkoduotus paketus.

Baito lygio RLE schemeje koduojamos identiškų baitų reikšmės, nepaisant nuskaitytoje eilutėje esančių atskirų bitų ar žodžių ribų. Labiausiai paplitęs baito lygio RLE schemas potipis yra baitų sekų kodavimas, sudarant dviejų baitų RLE paketus. Pirmasis baitas pakete yra sekos ilgio skaitliukas (reikšmė nuo 0 iki 255), o sekantis baitas žymi sekos simbolio reikšmę (1.33 pav., b). Šią

schema taip pat galima naudoti efektyviam nekoduotos šaltinio informacijos duomenims koduotų duomenų sraute laikyti (saugoti).

Tokioje modifikuotoje 2 baitų RLE paketo scheme, paskutiniai septyni mažiau svarbūs pirmojo baito bitai žymi sekos ilgį minus vienetas. Svarbiausio pirmojo baito bito reikšmė nusakomas sekos tipas (1.33 pav., c dalis). Jei svarbiausio bito reikšmė yra lygi 1-am, tai indikuoja, jog seka yra užkoduota (pavaizduota 1.33 pav., c dalyje). Koduotos sekos yra dekoduojamos nuskaitant sekos reikšmę, ją pakartojant tiek kartų, kokia yra sekos skaitliuko reikšmė. Jei svarbiausio bito reikšmė yra lygi 0-iui, indikuojama jog seka yra paraidinė, taip nurodant, jog po to sekantys baitai nuskaitomi nuosekliai, jų nedekoduojant. Nekoduotų baitų skaičius nurodomas RLE paketo sekos ilgio skaitliuke minus vienetas (pirmojo RLE paketo baito paskutiniai 7 bitai- intervalas nuo 0 iki 127, paveikslėlis 1.33 pav., d dalis). Baito lygio schemas gerai veikia, jei paveikslėlyje pikselio reikšmei koduoti skiriamas vienas baitas.

Pikselio lygio RLE schema naudojama, kuomet vienas paskui kitą einantys baitai yra naudojami vieno pikselio duomenims saugoti. Pikselio lygyje, bitai yra ignoruojami, o baitai skaičiuojami tik tam, kad identifikuotų pikselio reikšmę. Koduoto paketo dydis labai priklauso nuo koduojamo pikselio reikšmės. Pikselio informacijai saugoti sunaudotas bitų ar baitų kiekis yra nurodomas paveikslėlio failo antraštėje. Paveikslėlio pikselį identifikuojančių 3-jų baitų sekos, RLE koduojamos į 4-ių baitų paketą. Pirmasis (RLE pridedamas) paketo baitas nusako pasikartojančių baitų sekos ilgį (skaitliukas), sekantys trys yra sekos reikšmė (pavaizduota paveikslėlyje 1.33 pav., e dalis). Toks metodas iš esmės toks pats kaip ir baito lygio RLE, skiriasi tik paketo dydis.



1.33 pav. Bitų, baitų ir pikselio lygio RLE schemas.

Taip pat galima realizuoti schemą, kurioje paradinė nepasikartojančių pikselių seka, būtų įrašoma nekoduota. Veikimas pagrįstas jau anksčiau aprašytoje baitų lygio RLE schemoje, kuomet pirmasis baitų bitas yra naudojamas identifikuoti ar seka koduota ar ne. Svarbu nepamiršti, jog pikselio lygio RLE schemoje sekos ilgio skaitliukas nurodo tokių pačių *pikselių*, bet ne *baitų* kiekį.

Egzistuoja ir daugiau RLE modifikacijų, tokių kaip vertikalaus eilučių atsikartojimo kodavimas, tačiau šiame darbe tokios schemas yra neaktualios, todėl jų plačiau nenagrinėsime.

1.12. Sudaromo A&A algoritmo naudotojų analizė

Kuriamas algoritmas bus sudaromas taip, kad įgalintų jį panaudoti plačiau, nei vien JPEG standarte. Panašūs entropiniai algoritmai naudojami tiek daugelyje vaizdo failų formatų, tiek tekstinių duomenų sekų ilgio kodavimui. Taigi ir kuriamo algoritmo panaudojimo sritys išlieka tos pačios, tačiau neatmetama galimybė jį panaudoti ir kitur. Tai galėtų būti tam tikrų signalų duomenų kodavimas. Galimus algoritmo naudotojus pagal tipus, skirstytume į:

- Įvairių tipų failų formatų, kurie, kaip šaltiniai pasižymi neaukštu entropijos lygiu, kūrėjai ar savininkai;
- Programinės įrangos kūrėjai;
- Paprasti namų naudotojai, turintys programavimo žinių ir galintys realizuoti ir savo poreikiams naudoti A&A algoritmą.

1.13. Esamų kodavimo pertekliško mažinimo algoritmų analizė

Ankstesniuose skyriuose 1.9 Hafmano kodavimo algoritmas, 1.10 LZW kodavimo algoritmas ir 1.11 Run-Length Encoding algoritmas, apžvelgėme populiarius entropinio kodavimo algoritmus. Jie remiasi skirtingais principais, bet atlieka tą pačią funkciją- mažina kodavimo perteklišumą (žr. 1.4.2 Kodavimo pertekliškumas). Dauguma tokių algoritmų sugalvoti ir realizuoti jau seniai, štai Hafmano kodavimas aprašytas dar 1952 metais. Tuomet kompiuterių, kaip mes juos suprantame dabar, nebuvo. LZW prototipas LZ77 buvo realizuotas 1977 metais. Istorija ilga, tačiau jie sėkmingai adaptuoti kompiuterijoje bei naudojami ir dabar. Atsižvelgiant į minėtų algoritmų veikimo principus, gerąsias ir blogąsias savybes, sudaryta palyginamoji analizė, pateikta 1.3 lent.

1.3 lentelė Esamų entropinių kodavimo algoritmų palyginimas

Palyginimo kriterijus	Hafmano	LZW	RLE
Efektyviam kodavimui būtinos ilgos vienodų simbolių sekos	Ne	Ne	Taip
Koduojamos iš atsitiktinių simbolių susidariusios identiškios sekos	Taip	Taip	Ne
Kodavimas paremtas sekų pasirodymo tikimybe	Taip	Ne	Ne
Koduojamos sekos iš vienodų simbolių	Ne	Ne	Taip
Randamos įmanomai ilgesnės sekos ir koduojamos trumpesniu kodu	Taip	Taip	Ne
Veikimas	Vidutinis	Vidutinis	Greitas
Gali padidinti koduojamų duomenų apimtį	Ne	Ne	Taip

1.14. A&A algoritmas

Kaip minėta 1.11 skyriuje, RLE veikimo principas- sumažinti fizinį pasikartojančių simbolių sekos dydį. Šie pasikartojantys simboliai, kitaip- seka, koduojama dviem baitais (*angl. Byte*). Pirmasis baitas nusako pasikartojančių simbolių sekos ilgį ir yra vadinamas sekos skaitliuku. Antrasis baitas yra skaičiaus, iš kurių sudaryta seka, reikšmė ir jis vadinamas sekos reikšme. Iš 1.11 skyriuje pateiktų sekų pavyzdžių žinome, jog RLE efektyvumas pasiekiamas tada, kai yra ilgos pasikartojančių simbolių sekos. Jei tokių sekų nėra ar jos sąlyginai trumpos, RLE panaudojimas gali tapti ne tik kad neefektyvus, bet ir turėti neigiamą poveikį t.y. padidinti koduojamų duomenų apimtį. Norint efektyviai išnaudoti RLE privalumus ir pašalinti trūkumus reikia modifikuoti RLE schemą taip, kad nereikėtų naudoti RLE paketų ir būtų koduojamos tik pasikartojančių simbolių sekos. Nepasikartojantys simboliai būtų tiesiog perrašomi į išvedimo srautą, kaip koduota seka. Tokiu atveju, niekada nepadidėtų koduojamų duomenų apimtis, o esant sekoms, jos būtų efektyviai koduojamos. Išvardintais principais remiasi siūlomas naujas sekos ilgio kodavimo algoritmas, pavadinimu A&A.

1.15. Skaitmeninio vaizdo ir jo glaudinimo metodų analizės išvados

Skaitmeninių vaizdų glaudinimo algoritmų yra įvairių (JPEG, BTC, JPEG2000, JBIG). Vaizdų glaudinimui jie naudoja skirtingus principus, tačiau siektinas rezultatas visų tas patmaksimali vaizdo kokybė ir minimalus bylos dydis. Dėl savo privalumų JPEG yra populiariausias

vaizdų glaudinimo algoritmas, nors ir turintis nemažai trūkumų. Jis pasirinktas šio darbo šaltiniu, nes jo veikimas yra gana paprastas, o po sugludinimo gaunamo duomenų srauto struktūra nuspėjama ir dalinai kontroliuojama, taikant skirtingas kvantavimo matricas. Norint įvertinti kaip efektyviai galima užkoduoti šaltinio duomenis, reikia palyginti kelis panašaus tipo algoritmus.

Iš pateiktos lentelės (1.3 lent.) , matome, jog skirtingai nei aritmetiniai [16] ar žodyniniai entropijos, RLE tipo algoritmai nepasižymi itin aukštu naudingumu, tačiau jie paprasti realizacijos atžvilgiu, o bet to greitai veikia. Pastaroji savybė yra svarbi siekiant šio darbo tikslo. Kuriant A&A algoritmą, bus atsižvelgta į išvardintas gerąsias RLE tipo algoritmų savybes. Be kita ko bus pašalintas pagrindinis RLE trūkumas, kuriuo nepasižymi kiti du Hafmano ir LZW algoritmai- neigiamo kodavimo rezultato, kuomet šaltinis pasižymi aukštu entropijos lygiu. Jei įgyvendinus tikslus, bus pasiektas teigiamas rezultatas, A&A algoritmas bus integruotas į JPEG standartą su galimybe naudoti vietoj RLE.

2. EKSPERIMENTINĖS PROGRAMINĖS ĮRANGOS REIKALAVIMŲ SPECIFIKACIJA IR A&A ALGORITMO APRAŠAS

2.1. Eksperimentinė programinė įranga

RLE ir A&A algoritmų glaudinimo efektyvumui palyginti yra sukurta eksperimentinė programinė įranga. Šios programinės įrangos pagalba galima nusiskaityti pavyzdinių paveikslukų aibę ir užkoduoti ją JPEG algoritmu, duomenų sekoms suglaudinti naudojant tiek RLE, tiek ir A&A metodus. Suglaudinus paveikslukus vartotojui išvedama spalvų bei intensyvumo komponenčių ($Y' C_B C_R$) baitų sekų ilgiai, gauti suglaudinus jas kiekvienu iš minėtų metodų. Palyginamoji analizė vykdoma neglaudintų baitų sekų ilgio atžvilgiu. Rezultatai analizuojami Microsoft Excel programinės įrangos priemonėmis. Programinės įrangos kodas pateiktas 6.1 priedas. Programinės įrangos kodas.

2.2. Funkciniai ir nefunkciniai reikalavimai

Eksperimentinė programinė įranga pilnai atitinka šiuos **funkcinius reikalavimus**:

- Turi būti galimybė nusiskaityti nustatytą kiekį testinių paveikslukų;
- Nuskaityti paveikslukai turi būti suglaudunami pagal JPEG standartą;
- Nuskaityti paveikslukai turi būti pakartotinai užkoduojami pagal visus JPEG standarto reikalavimus, tačiau kvantuota skaičių seka turi būti suglaudinama dviem RLE ir A&A metodu.
- Programinė įranga turi vartotojui išvesti kiekvienos iš spalvų bei intensyvumo komponenčių neglaudintus baitų sekos ilgį, suglaudintus dviem RLE ir A&A metodu.
- Programinė įranga turi vartotojui išvesti kiekvienos iš spalvų bei intensyvumo komponenčių baitų sekos ilgį, suglaudintus dviem RLE ir A&A metodu.
- Programinė įranga turi apskaičiuoti ir vartotojui išvesti viso suglaudinto paveiksluko užimamą vietą naudojantis kiekvienu testuojamu metodu, nesudarant failo struktūros ir nepriedant metaduomenų.

Eksperimentinė programinė įranga atitinka šiuos **nefunkcinius reikalavimus**:

- Turi būti palaikomi tiek spalvoti, tiek ir nespalvoti paveikslukai.

2.3. Algoritmo sudarymas

Skyriuje 1.11 Run-Length Encoding algoritmas, pateiktame pavyzdyje (1.30 pav.), apskaičiavome, jog simbolių sekos „ABCDEFGHIIJKLM“ ilgis, koduojant RLE, duomenų apimtį padidino du kart, nes kiekvieno baito dydžio simboliui užrašyti sunaudoti du baitai. Kaip galima išvengti neigiamo tokio glaudinimo rezultato? Tai išanalizuosime toliau darbe.

Siūlomam algoritmui realizuoti pakaktų pažymėti tą vietą, kurioje yra seka. Sekoje pasikartojančių simbolių (baitų) kiekis turėtų būti ne mažesnis nei tokiu būdu kodavimui sunaudojamų simbolių (baitų) kiekis. Kitaip algoritmas bus neefektyvus arba turės neigiamą rezultatą.

Algoritmo struktūra

Kadangi JPEG standarte 8x8 bloko elemento kiekvienai $Y' C_B C_R$ komponentei koduoti naudojamas baitas, tai ir sudarytas simbolių sekas A&A algoritmas nuskaito po vieną baitą (simbolis- 8 bitai). Iš viso to galima daryti išvadą, jog sekos žymeklis taip pat turėtų būti vieno baito apimties.

Keleto pasikartojančių simbolių sekai pažymėti jau sunaudojamas vienas baitas, dar neaprašius koduojamų simbolių. Sekai tinkamai užkoduoti, kaip ir RLE, reikės naudoti tiek pasikartojančio simbolio reikšmę, tiek ir pasikartojimų skaičių nurodantį skaitliuką. Taigi viso bus reikalingi trys baitai.

Algoritmo veikimo principas

Algoritmas, koduodamas seką ir joje radęs pasikartojančių simbolių, įterpia žymeklį toje vietoje, kuri seka po pirmojo pasikartojančio simbolio. Simbolis, esantis prieš žymeklį, lieka nepakitęs ir nurodo *sekos reikšmę*. Po žymeklio įrašomas *sekos skaitliukas* t.y. skaičius, nurodantis kiek kartų pasikartoja sekos reikšmė.

Dekodavimo proceso metu, algoritmas nuskaitytinėja koduotos sekos elementus (pažymėkime juos i – taisiais elementais, $1 \leq i \leq n$) tol, kol i elementas yra lygus žymekliui ($i ==$ "Žymeklis"). Radęs žymeklį, algoritmas prieš elementą $i ==$ "Žymeklis", esantį simbolį ($i - 1$) pakartoja tiek kartų, kiek yra nurodyta po žymeklio ($i + 1$) einančiu skaičiumi. Gautas rezultatas įrašomas į išvedimo seką. Tuomet vykdomas sekantis žingsnis, nuskaitytas po *sekos skaitliuko* einantis elementas ($i + 2$).

Algoritmo efektyvumas

Panagrinėkime 11-os skaičių seką (2.1 pav.3.1 pav.), kurioje pasikartojimų skaičius (pažymėkime jį k) yra mažesnis arba lygus trims ($k \leq 3$):

15 16 19 25 25 25 3 8 5 5 13

2.1 pav. Nekoduota 11-os skaičių seka

Tuomet, panaudojus A&A algoritmą, ši seka būtų užkoduota taip (2.2 pav.3.2 pav.):

15 16 19 25 "ŽYMEKLIS" 3 9 8 5 "ŽYMEKLIS" 2 13

2.2 pav. A&A algoritmu užkoduota 11-os skaičių seka

Iš pateikto pavyzdžio matyti, jog tada kai $k = 3$, A&A algoritmu užkoduota seka užima 100% nekoduotos sekos duomenų apimtį. Jei $k = 2$, algoritmas, įterpdamas žymeklį, sunaudojo papildomą baitą. Iš pateiktos informacijos daroma išvada, kad norint maksimaliai supaprastinti ir taip pagreitinti algoritmo kodavimo ir dekodavimo procesus, bei pasiekti naudingumą, kuris viršytų 100%, reikia, kad sekoje būtų bent keturi pasikartojantys simboliai t.y. tada, kai $n \geq 3$. Jei $n \leq 3$, seka nekoduojama ir tiesiog perrašoma į išvedimo seką.

Pagrindiniai algoritmo žingsniai

Kodavimas:

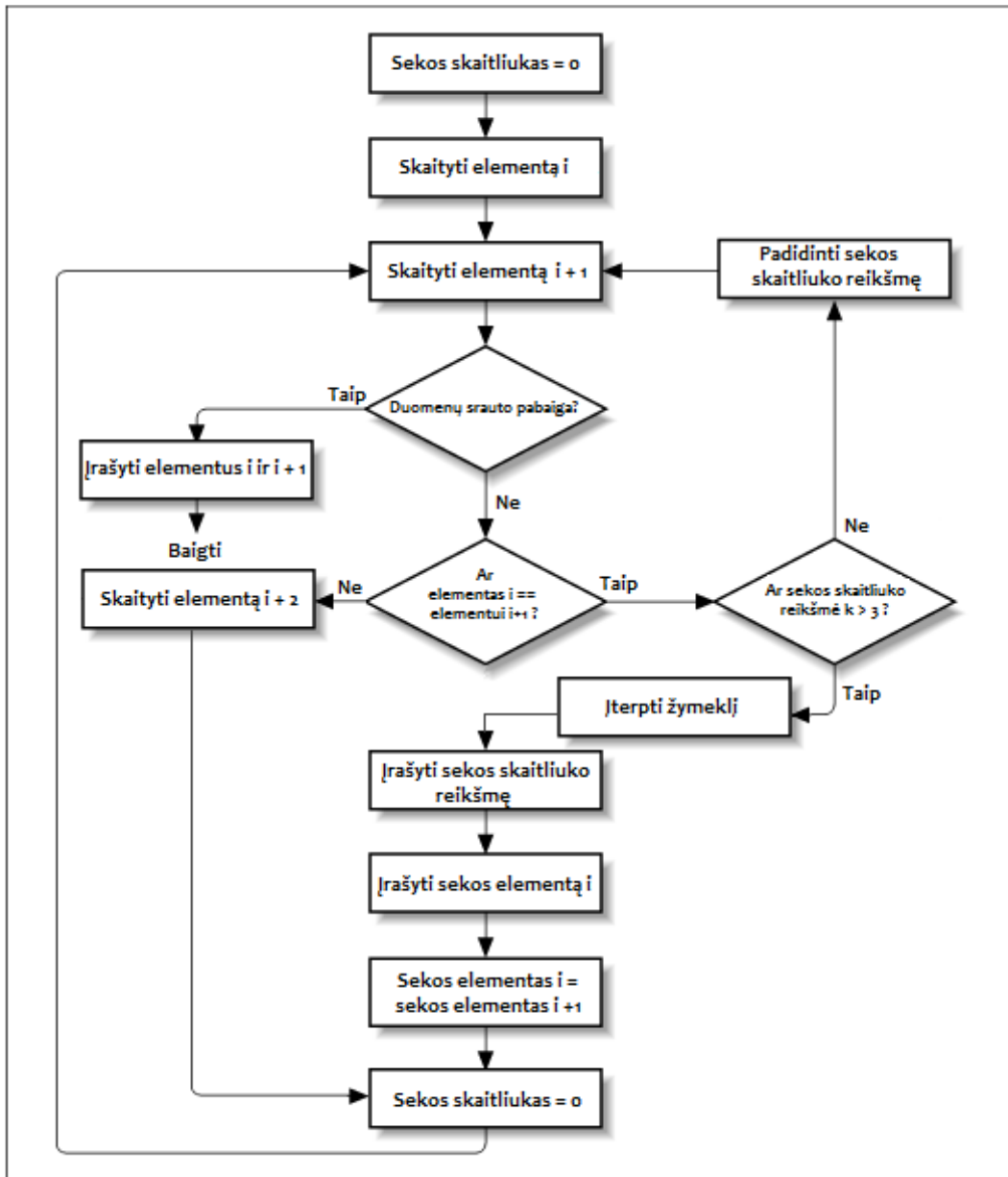
1. Įėjimo srautas yra JPEG Y'CB_R spalvų komponentų kvantuotų matricų elementai ($1 \leq i \leq n$), nuskaityti zigzago principu (vienmatis masyvas).
2. Elementai nuskaityti nuosekliai, tikrinama ar dabartinis elementas ($i = 1$) lygus prieš tai buvusiam ($i == i - 1$);
3. Jei taip, imamas sekantis elementas ir tikrinama ar jis lygus prieš tai buvusiam. Procesas kartojamas, kol pakinta sekos elementas ($i \neq i - 1$) arba pasikartojimų skaičius mažesnis arba lygus trims ($k \leq 3$).
4. Jei sekos elementas pakinta ($i \neq i - 1$), bet prieš tai buvusio elemento pasikartojimų skaičius yra didesnis už tris ($k > 3$), algoritmas įterpia žymeklį po pasikartojančio elemento, o sekos skaitliuką įrašo po žymeklio. Tokia seka įrašoma į išvedimo srautą.
5. Jei sekos elementas pakinta ($i \neq i - 1$) ir prieš tai buvusio elemento pasikartojimų skaičius yra mažesnis arba lygus trims ($k \leq 3$), algoritmas tokią seką perrašo tiesiai į išvedimo srautą, jos nekoduodamas.
6. . Paskutiniai trys sekos elementai ($i = n - 2$, $i = n - 1$ ir n) nekoduojami, perrašomi tiesiogiai į išvedimo srautą. Algoritmas baigia darbą, kai pasibaigia sekos elementai ($i = n$).

Dekodavimas:

7. Įėjimo srautas yra A&A algoritmo koduota elementų seka (vienmatis masyvas).

8. Elementai nuskaitymi nuosekliai ir perrašinėjami tiesiogiai į išvedimo srautą tol, kol dabartinis elementas lygus žymekliui ($i == \text{"Žymeklis"}$).
9. Jei taip, tai prieš tai buvęs elementas ($i - 1$) pakartojamas tiek kartų, kiek yra nurodyta po žymeklio esančiu skaičiumi.
10. Vykdomas 2 žingsnis, kol pasibaigia sekos elementai ($i = n$).

Žemiau pateikiama (2.3 pav.) A&A algoritmo sekų diagrama. Jos struktūra panaši į RLE algoritmo, tačiau skiriasi keli elementai, kurių dėka įgyvendinamas principas koduoti tik tuomet, kai yra seka ir tik tas sekas, kurių elemento pasikartojimų skaičius didesnis už tris kartus.



2.3 pav. A&A algoritmo sekų diagrama

Algoritmo taikymas

Pagrindinis šio algoritmo tikslas- panaudojimas JPEG standarte, pakeičiant šiuo metu jame naudojamą RLE algoritmo modifikaciją, kuri koduoja po kvantavimo 8x8 blokuose susidarančiose sekose itin dažną skaičių „0“ (žr. 1.7.2.4 posk.). Kadangi JPEG atveju, koduojami simboliai yra skaitiniai, tai žymekliui racionaliausiai naudoti raidinį simbolį (tarkime „A“ raidę). Ir atvirkščiai- jei

koduojamos sekos yra raidinės, tuomet geriau naudoti skaitinį žymeklį (tarkime „1“). Taip bus išvengta galimų veikimo trukdžių. Jei žymekliui naudotume simbolį, esantį koduojamų simbolių intervale, tuomet reikėtų papildomai įdėti saugiklį. Šis saugiklis garantuotų, jog kodavimo algoritmas, kiekvieną kartą sutikęs žymeklio reikšmę atitinkantį simbolį, prieš jį įrašytų sekos reikšmę (šiuo atveju ji lygi žymekliui), o po to simbolio, įrašytų sekos skaitliuką. Taip užkoduota seka, būtų tinkamai dekoduojama, tačiau jei žymeklio reikšmę atitinkančių simbolių pasikartojimų skaičius būtų mažesnis už tris, kodavimas taptų neefektyvus. Būtų sunaudojami papildomi baitai- A&A algoritmas netektų savo privalumo niekada nepadidinti koduojamų duomenų apimties.

Jei nekoduotų duomenų sraute pasitaiko simbolis, kuris yra lygus žymekliui, o algoritme nenaudojamas saugiklis, tuomet prieš jį einantis simbolis bus pakartotas tiek kartų, kiek nurodo už jo esantis skaičius (žr. šio sk. Algoritmo veikimo principas). Taigi tokiu atveju, nebus įmanoma atkurti užkoduotų duomenų, nes kodavimo algoritmas juos negrįžtamai iškraipys. Visgi labai paprasta adaptuoti algoritmą bet kokiais sričiais, tiesiog naudojant žymeklį, kuris koduojamame duomenų sraute visiškai nepasitaiko.

JPEG kvantuotų blokų elementų reikšmės kinta skaičių intervale nuo -127 iki 128 (žr. 4 DCT skaičiavimas), taigi naudojant raidinį žymeklį- simbolį „A“, yra garantuojama, kad toks simbolis sekoje tikrai nepasitaikys.

JPEG 8x8 bloko elementų skaičius yra 64. Kadangi blokai koduojami atskirai (žr. 3 Skaidymas į blokus), tai naudojant A&A algoritmą JPEG kvantuotų blokų elementų sekoms koduoti, įėjimo sekos elementų skaičius visada žinomas ir yra lygus 64. Toks baigtinis ir iš anksto žinomas šaltinio elementų skaičius garantuoja, jog sekos ilgio skaitliuko visada užteks tos sekos ilgiui užkoduoti (skaitliukas yra 1 baito dydžio, tai viso $2^8 = 256$ reikšmės, o pakanka tik 64).

A&A algoritmo palyginimas su RLE

JPEG standarto algoritmas yra greitai ir efektyviai veikiantis, todėl galima būtų manyti, jog tobulinti nėra ką. Tačiau kaip JPEG standarto dalis, jame naudojamas ir kitas algoritmas- RLE. Šio darbo tikslui įgyvendinti pasirinktas jo pakeitimas efektyvesniu to paties tipo algoritmu. Tam, kad teoriniu aspektu, būtų galima objektyviai įvertinti A&A algoritmo panaudojimo tikslingumą, reiktų išskirti abiejų algoritmų privalumus ir trūkumus ir juos tarpusavyje palyginti. Vėliau ši palyginamoji analizė bus atlikta praktikoje, realizuojant abu algoritmus pasirinktoje „MATLAB“ programavimo kalboje. Palyginimui taip pat bus naudojamas ir standartinis RLE baito lygio variantas (žr. 1.11 sk.). Trumpai aptarkime kiekvieną iš algoritmų ir išskirkime jų privalumus ir trūkumus.

Standartinis RLE (angl. Run-length encoding) duomenų sekos ilgio kodavimo algoritmo variantas, kodavimui naudojantis dviejų baitų paketus, gerai koduoja bet kokių pasikartojančių skaičių sekas, ir gerai veikia jei tik tos sekos dažnos ir pakankamai ilgos t.y. sekų kodavimo metu sutaupyti baitai, turėtų kompensuoti praradimus, atsirandančius kuomet vieno simbolio seka, užrašoma į dviejų baitų paketą. Pirmasis paketo simbolis žymi pasikartojimų skaičių, antrasis pasikartojimų reikšmę. Pavyzdžiui seka [4; 0; 0; 3; 0; 0; 0; 0; 2; 0; 0; 0] standartiniu RLE, bus užkoduotai kaip [(0, 4); (2, 0); (0, 3); (4, 0); (0, 2); (3, 0)]. Kaip jau minėta 1.11 skyriuje, jis gerai koduoja binarinius, tokie kaip knygos puslapiai, vaizdus. Nemažas efektyvumas pasiekiamas ir koduojant tekstūrų paveikslėlius, nes juose yra daug atsikartojimų- vienerūšių duomenų. Trūkumai pasireiškia tuomet, kai vaizdai yra spalviškai įvairūs, sudėtingi, juose daug kontrastų. Tokių šaltinių generuojamose duomenų srautuose, vienerūšių duomenų atsikartojimai reti, todėl ir algoritmo efektyvumas nedidelis [12]. Tokio algoritmo kodas MATLAB kalba:

```

function output = rle(input)
% seq –šaltinio seka
% l –šaltinio sekos ilgis
% j –išvedimo sekos elementas
% i –šaltinio sekos elementas
seq = 0;
j = 1;
l = length(input);
for i = 1 : l
    if (input(i) == 0)
        seq = seq + 1;
    else
        output(j) = seq;
        output(j + 1) = input(i);
        j = j + 2;
        seq = 0;
    end;
end;
if (seq > 0)
    output(j) = 0;

```

Modifikuota RLE versija, koduojanti tik nulius. Ši modifikacija adaptuota JPEG standartui, nes šis skaičius šaltinio sraute labai dažnas (žr. sk. 1.7.2.4). Šis algoritmas taip pat naudoja dviejų baitų paketus, tik skirtingai nei standartinėje RLE modifikacijoje, pirmasis paketo skaičius žymi kiek „0“-ių yra iki skaičiaus, o antrasis yra pats skaičius. Pora (0, 0) nurodo bloko pabaigą [2]. Ši pora pažymi vietą, nuo kurios iki pat sekos pabaigos nuosekliai einantys elementai yra tik skaičiai „0“. Pavyzdžiui anksčiau pateikta seka [4; 0; 0; 3; 0; 0; 0; 0; 2; 0; 0; 0] modifikuotu RLE, bus užkoduota kaip [(0, 4); (2, 3); (4, 2); (0, 0)]. Savaiame suprantama toks glaudinimas bus efektyvus tik tokiuose sekose, kur palyginus su kitais skaičiais, „0“ yra itin dažnas ir sudarantis vientisas sekas. Tokio algoritmo kodas MATLAB kalba.

```

function output = rle_mod(input)
% seq –šaltinio seka
% l –šaltinio sekos ilgis
% j –išvedimo sekos elementas
% i –šaltinio sekos elementas
seq = 1;
j = 1;
l = length(input);
last = input(1);

for i = 2 : l
    if (input(i) == last)
        seq = seq + 1;
    else
        output(j) = last;
        output(j + 1) = seq;
        j = j + 2;
        seq = 1;
        last = input(i);
    end;
end;
output(j) = last;
output(j + 1) = seq;

```

Jau žinome, kad tokia RLE algoritmo realizacija visas sekas koduoja, jas sudedant į dviejų baitų paketus. Taigi bet koku atveju- ar koduojant bet kokius skaičius (RLE), ar tik nulius (modifikuota RLE), pavienių simbolių sekos bus koduojamos neefektyviai. Taip pat žinome, kad ilgos „0“ sekos susidaro tik didinant suspaudimo laipsnį. Jei suspaudimas mažesnis, paliekamos aukštadažnės spalvų dedamosios (žr. 5 Kvantavimas), tuomet jų reikšmės yra ne „0“. Taigi nei vienas iš aukščiau išvardintų algoritmų nėra universalus ir savo aukščiausią efektyvumą pasiekia tik esant tam tikroms sąlygoms.

Jei norėtume išsaugoti gerą vaizdo kokybę, tačiau sumažinti vaizdo failo apimtį, reiktų naudoti mažą suspaudimo laipsnį. Tokiu atveju pats JPEG algoritmas veiktų efektyviai, dėl savo savybių ir privalumų, išvardintų 1.7.2.3 skyriuje JPEG privalumai ir trūkumai. Tačiau RLE algoritmo variantai veiktų neefektyviai, dėl trumpų pasikartojančių skaičių sekų. Ypač tai aktualu, kai koduojamas vaizdas yra labai spalvotas ir kontrastingas.

Sekos kodavimas standartiniu RLE. Seka būtų užkoduota taip: [(0, -23); (0, -3); (0, 0); (0, -3); (0, -2); (0, -6); (0, 2); (0, -4); (0, 1); (0, -3); (2, 1); (0, 5); (0, 1); (0, 2); (0, -1); (0, 1); (0, -1); (0, 2); (5, 0); (2, -1); (38, 0)]. Suskaičiavus yra 21 paketas po 2 baitus, taigi 42 baitai vietoje 64-ių. Panaudojus glaudinimo lygio apskaičiavimo (5) formulę nurodytą 1.6.1 skyriuje, galima įvertinti RLE glaudinimo lygį:

$$\alpha_{RLE} = \frac{I}{I'} = \frac{64}{42} \approx 1,52$$

Koeficientas α_{RLE} parodo, jog koduoti duomenys, užima maždaug 1,52 karto mažiau vietos, nei neglaudinti duomenys.

Sekos kodavimas modifikuotu RLE. Seka būtų užkoduota taip: [(0, -23); (0, -3); (1, -3); (0, -2); (0, -6); (0, 2); (0, -4); (0, 1); (0, -3); (0, 1); (0, 1); (0, 5); (0, 1); (0, 2); (0, -1); (0, 1); (0, -1); (0, 2); (5, -1); (0, -1); (0, 0)]. Suskaičiavus, kaip ir standartiniu RLE yra 21 paketas po 2 baitus, taigi taip pat 42 baitai vietoje 64-ių ir glaudinimo lygis $\approx 1,52$

Šioje sekoje abu algoritmai buvo vienodai efektyvūs. Standartinis RLE buvo ne toks efektyvus glaudinant nulius, bet kitus pasikartojusius skaičius suglaudino efektyviau nei modifikuotas RLE.

Sekos kodavimas A&A. Naudojant A&A algoritmą, seka nebūtų užrašoma į paketus, koduojami tik skaičiai, pasikartojantys bent keturis kartus, taigi seka būtų užkoduota taip: [-23; -3; 0; -3; -2; -6; 2; -4; 1; -3; 1; 1; 5; 1; 2; -1; 1; -1; 2; 0; A; 5; -1; -1; 0; A; 38]. Suskaičiavus yra 27 simboliai (įskaitant ir žymeklius) po 1 baitą, taigi 27 baitai vietoje 64-ių. Panaudojus glaudinimo lygio apskaičiavimo (5) formulę, galima įvertinti RLE glaudinimo lygį:

$$\alpha_{A\&A} = \frac{I}{I'} = \frac{64}{27} \approx 2,37$$

Taigi koeficientas parodo, jog suglaudinti duomenys užims maždaug 2,37 karto mažiau vietos, nei neglaudinti duomenys. Nors pateiktoje sekoje buvo ir skaičių pasikartojimų ir vientisų skaičius „0“ sekų t.y. atitiko sąlygas, kuriomis gerai veikia abu RLE algoritmai, palyginus standartinio ir modifikuoto RLE glaudinimo lygius (šiuo atveju jie vienodi), su A&A algoritmo lygiu ($\frac{I}{I'} = \frac{2,37}{1,52} \approx 1,55$), gauname rezultatą, kuris parodo, jog A&A algoritmas maždaug 1,55 karto geriau suglaudino duotos sekos duomenis. Galima daryti išvadą, jog A&A algoritmas yra universalus ir parankesnis naudoti JPEG standarte, nei kuris nors RLE algoritmo variantas.

Sekančiame skyriuje bus apžvelgta ir įgyvendinta praktinė JPEG standarto realizacija MATLAB kalboje. Taip pat bus realizuoti ir visi trys analizuojami entropinio glaudinimo algoritmai: RLE, jo JPEG modifikacija ir A&A algoritmas. Bandant pagrįsti anksčiau šiame skyriuje išdėstytas teorines prielaidas ir patvirtinti priimtas išvadas, visų šių algoritmų efektyvumas bus patikrintas praktiniais bandymais, tam naudojant vaizdo ir jo fragmentų duomenų bazę.

3. A&A ALGORITMO REALIZACIJA IR TYRIMAS

3.1. Realizacijos modelis

Šiame skyriuje pateikiamas aprašymas, kaip buvo vykdomi bandymai, naudojant eksperimentinę programinę įrangą, kaip buvo pasirinkti paveikslėliai ir ką šiais bandymais buvo siekiama parodyti.

3.2. Pradiniai duomenys

Kvantavimo rezultatų duomenų glaudinimo algoritmams patikrinti buvo pasirinkta 100 įvairaus dydžio ir spalvingumo paveikslėlių. Šis skaičius pasirinktas taip, kad rezultatai būtų lengvai ir paprastai suprantami ir interpretuojami.

Paveikslėlių bazė buvo sudaryta atsižvelgiant į eksperimentui naudojamų algoritmų efektyvumo specifiką. Standartinis RLE algoritmas gerai glaudina tekstūros paveikslėlių generuojamas sekas, todėl 20 iš 100 paveikslėlių bus tokio tipo. Šis algoritmas turėtų gerai glaudinti ir binarinių paveikslėlių duomenų sekas, tam patikrinti pasirinkta naudoti 20 juodai baltų paveikslėlių, iš kurių 10 bus atsitiktinai parinktų knygų puslapiai. Dar 20 paveikslėlių bus natūralios gamtos vaizdų, nes standartinis RLE čia turėtų neparodyti didelio efektyvumo, o RLE modifikacija, skirta JPEG algoritmo kvantavimo rezultatų duomenims glaudinti, turėtų būti pranašesnė. 20 grafikų ir diagramų paveikslėlių pasirinkta norint įvertinti ar galima rekomenduoti naudoti analizuojamus algoritmus tokio tipo paveikslėlių generuojamoms duomenų sekoms glaudinti, nes literatūroje apie tai mažai užsimenama. Paskutiniai 20 paveikslėlių bus didmiesčių vaizdai. Juose turėtų būti ir pakankamai daug kontrastų, ir kitavertus, daug atsikartojančių tekstūrų ir vienodų spalvų. 3.1 lent. pateikta susisteminta informacija iš kokių tipų sudaryta paveikslėlių bazė ir kiekvieno tipo imtis.

3.1 lentelė Analizuojamų paveikslėlių kiekis ir tipai

Paveikslėlių tipas	Kiekis, vnt.	Eiliškumas (imtiniai) vykdant bandymą
Tekstūriniai	20	1-20
Juodai balti	10	21-30
Knygų puslapiai	10	31-40
Natūralios gamtos	20	41-60
Grafikai ir diagramos	20	61-80
Didmiesčiai	20	81-100

Naudojantis sukurta eksperimentine programine įranga buvo testuojama kokio dydžio bylos gaunamos suglaudinus juos JPEG algoritmu, kvantavimo rezultatus koduojant RLE ir A&A algoritmais. Čia byla yra duomenų apimtis, reikalinga suglaudintiems duomenims laikyti. Galutinė failo struktūra nebuvo sudaroma ir metaduomenys nepriedami.

Sudarant paveikslėlių duomenų bazę nebuvo atsižvelgta į tai, ar JPEG algoritmas yra rekomenduotinas glaudinti tokius vaizdus. Aktualia sritimi čia laikomos tik šaltinio sugeneruotos sekos.

3.3. Algoritmų veikimo patikra

Atliekant bandymus, rezultatai turėtų būti tikslūs, neginčijami, o jų vertinimas turi būti objektyvus. Norint patikrinti ar visi algoritmai realizuoti atitinkamai jų specifikacijoms, parinktas atsitiktinis 16x16 pikselių išmatavimo paveikslėlis (3.1 pav.). Toks dydis parinktas specialiai tam, kad gautų sekų skaičius būtų nedidelis ir būtų paprasta palyginti ar nekoduota seka visais trimis algoritmais užkoduojama tinkamai t.y. taip kaip aprašyta tų algoritmų specifikacijoje, tačiau kol kas

koduotas_RLE =																			
Columns 1 through 20																			
-3	1	-6	1	0	1	3	1	4	1	-2	1	-3	1	1	1	2	1	-2	1
Columns 21 through 40																			
1	1	-1	2	0	3	1	2	-1	1	0	1	1	1	0	1	1	1	41	0

3.2 pav. Programinės įrangos išvesties kodas, RLE koduota seka

Tai pačiai sekai užkoduoti, modifikuota RLE versija, pakako tik 36 baitų arba 18-os dviejų baitų paketų. Tarpiniai modifikuoto RLE algoritmo išvesties duomenys pateikti 3.3 pav.

koduotas_RLE_MOD =																			
Columns 1 through 20																			
0	-3	0	-6	1	3	0	4	0	-2	0	-3	0	1	0	2	0	-2	0	1
Columns 21 through 36																			
0	-1	0	-1	3	1	0	1	0	-1	1	1	1	1	0	0				

3.3 pav. Programinės įrangos išvesties kodas, modifikuotu RLE koduota seka

A&A algoritmas sunaudojo tik 26 baitus duotai sekai užkoduoti (3.4 pav.).

koduotas_AA =																			
Columns 1 through 20																			
-3	-6	0	3	4	-2	-3	1	2	-2	1	-1	-1	0	0	0	1	1	-1	0
Columns 21 through 26																			
1	0	1	0	A	41														

3.4 pav. Programinės įrangos išvesties kodas, A&A koduota seka

Šis algoritmų patikrinimas parodė, jog duotai sekai koduoti, efektyviausias yra A&A algoritmas. Jis sunaudojo mažiausiai- 26 baitus. 36 baitus sunaudojo RLE JPEG modifikacija- tai antrasis pagal efektyvumą algoritmas. Standartinis RLE sunaudojo net 40 baitų ir tai yra prasčiausias rezultatas.

Žinoma vienos sekos nepakanka, kad būtų galima teigti, jog vienas ar kitas algoritmas yra visapusiškai geriausias ir efektyvus. Pabandykime nustatyti kokiomis sąlygomis A&A algoritmo efektyvumas mažesnis, dėl jo sunaudojamo papildomo baito žymekliui įterpti.

Panagrinėkime seką, kurios tik vienas elementas yra skaičius „-12“, o visi likę skaičiai yra „0“. Ši seka taip pat paimta iš eksperimentinės programinės įrangos išvesties kodo. Seka sugeneruota glaudinant 3.1 pav., tai spalvos komponentės matricos elementai, nuskaityti zigzago principu. Seka yra tokia: [-12; 0; ... ; 0], viso 64 elementai. Užkodavus gauname tokius kodus:

- RLE algoritmu: [1; -12; 63; 0];
- RLE modifikuotu algoritmu: [0; -12; 0; 0];
- A&A algoritmu: [-12; 0; A; 63].

Nors algoritmų veikimo principai skiriasi, šiuo atveju jie visi parodė vienodą efektyvumą, duotai sekai užkoduoti sunaudodami po 4 baitus. A&A algoritmo žymeklijo panaudojimas neturėjo neigiamos įtakos. Čia galima atkreipti dėmesį, koks didelis efektyvumas pasiekiamas koduojant sekos ilgį. Visi algoritmai sunaudojo net 16 kartų mažiau vietas.

JPEG standarte, kvantuojant spalvų komponentių C_R ir C_B matricas, paliekamos tik žemadažnės dedamosios (žr. 5 Kvantavimas), ir dauguma šių matricų elementų, po kvantavimo suapvalinami iki artimiausio sveiko skaičiaus, kuris dažniausiai yra „0“. Panagrinėkime seką, kurios visi 64 elementai yra skaičius „0“: [0; ... ; 0]. Iš tokių elementų susidarė ir analizuojamojo 3.1 pav. paveikslėlio duomenų seka. Naudojant programinę įrangą, algoritmai ją užkodavo taip:

- RLE algoritmas: [64; 0];

- RLE modifikuotu algoritmu: [0; 0];
- A&A algoritmu: [0; A; 64].

RLE algoritmas pažymėjo, kad 64 kartus atsikartoja skaičius „0“. Modifikuotas RLE įrašė sekos pabaigos žymeklį, kuris šiuo atveju reiškia, jog visi sekos elementai yra „0“. Šių algoritmų efektyvumas yra vienodas. Prasčiausiai pasirodė A&A algoritmas, nes jam prireikė įterpti baidinį žymeklį „A“, todėl vietoj galimų dviejų baitų, jis sunaudoja tris. Ši situacija ir vienintelė, kada A&A algoritmas buvo mažiau efektyvus, palyginus su kitais dviem RLE algoritmais. Tačiau tokios sekos turėtų būti pakankamai retos, todėl galima manyti, kad bendras A&A algoritmo naudingumas bus didesnis už kitų dviejų. Tam kad patvirtinti ar paneigti šią prielaidą, bus atliekamas eksperimentas, naudojant pagal klasterį sugrupuotus paveikslukus, aprašytus skyriuje 3.2 Pradiniai duomenys.

3.5. Rezultatai

Išvesties tekstas, gautas ištestavus algoritmus su eksperimentine programine įranga, pateiktas prieduose 6.2, 6.3 ir 6.4. Eksperimentą sudaro trys dalys. Jos skiriasi paveikslėlių glaudinimui naudojamais suspaudimo lygiais. Kiekvienoje dalyje bus analizuojami visi 100-as paveikslukų. Kvantavimo matricos dydis procentais, atvirkščiai atitinka glaudinimo lygį¹. Pirmoje dalyje bus pateikiami rezultatai, gauti naudojant standartinę 50% kvantavimo matricą. Pirmos dalies visi bandymų rezultatai bus pateikti lentelėje 3.2 lent. Antroje dalyje bus naudojama 70% kvantavimo matrica, o trečioje- 30%. Pilni pastarųjų dalių rezultatai lentelių pavidalu nebus pateikti. Informacija, gauta išanalizavus pirmos ir antros dalių rezultatus, bus susisteminta ir pateikta atskiromis lentelėmis kiekvienai daliai. Rezultatai išvedimi procentinė išraiška, palyginimui naudojant:

- Nesuglaudintų ir suglaudintų bylų duomenų apimties baitais;
- Algoritmų tarpusavio efektyvumo palyginimą.

3.5.1. Pirmos dalies bandymo rezultatai

Bylų, suglaudintų naudojant JPEG standarto 50% kvantavimo matricą, o susidariusioms sekos koduoti naudojant abu RLE bei A&A algoritmus, duomenų apimties rezultatai pateikti lentelėje- 3.2 lent. Programinės įrangos skaičiavimo rezultatai pateikti 6.2 priedas. Pirmos dalies bandymų rezultatai.

3.2 lentelė Bylų, suglaudintų JPEG standartu (glaudinimo lygis 50%), naudojant abu RLE algoritmus ir A&A algoritmą palyginimas

Bylos pavadinimas	Nesuglaudintos bylos dydis baitais	Bylos, suglaudintos RLE algoritmu, dydis baitais	Neglaudintų ir suglaudintų RLE algoritmu duomenų apimties santykis, %	Bylos, suglaudintos modifikuotu RLE algoritmu, dydis baitais	Neglaudintų ir suglaudintų modifikuotu RLE algoritmu duomenų apimties santykis, %	Bylos, suglaudintos A&A algoritmu, dydis baitais	Neglaudintų ir suglaudintų A&A algoritmu duomenų apimties santykis, %
Kvantavimo matrica 50%							
Tekstūrų paveikslėliai							
1 .jpg	147456	30498	20,68	24662	16,72	24706	16,75
2 .jpg	79800	18066	22,64	16698	20,92	14240	17,84
3 .jpg	79800	14814	18,56	12454	15,61	12590	15,78
4 .jpg	79800	19778	24,78	17606	22,06	15625	19,58
5 .jpg	79800	30610	38,36	27270	34,17	22299	27,94
6 .jpg	79800	9390	11,77	8204	10,28	8502	10,65
7 .jpg	79800	25342	31,76	22618	28,34	19169	24,02
8 .jpg	79800	13848	17,35	12388	15,52	11558	14,48
9 .jpg	79800	16226	20,33	14116	17,69	13697	17,16
10 .jpg	90000	14588	16,21	12460	13,84	12232	13,59

¹ Kuo kvantavimo matricos nuošimtis didesnis, tuo glaudinimo laipsnis yra mažesnis ir atvirkščiai.

11	.jpg	90000	11178	12,42	9546	10,61	9779	10,87
12	.jpg	90000	24552	27,28	21378	23,75	17959	19,95
13	.jpg	90000	48438	53,82	41824	46,47	34731	38,59
14	.jpg	90000	25552	28,39	20632	22,92	19459	21,62
15	.jpg	90000	28354	31,50	24158	26,84	20183	22,43
16	.jpg	90000	25394	28,22	22808	25,34	18071	20,08
17	.jpg	90000	21038	23,38	17456	19,40	16210	18,01
18	.jpg	5760000	1325644	23,01	1134216	19,69	989624	17,18
19	.jpg	5760000	1141748	19,82	984178	17,09	877269	15,23
20	.jpg	90000	35738	39,71	30682	34,09	25239	28,04
Dvispalviai paveikslėliai								
21	.jpg	3932160	429620	10,93	385274	9,80	380883	9,69
22	.jpg	3686400	385678	10,46	336714	9,13	347888	9,44
23	.jpg	708048	166158	23,47	146644	20,71	125213	17,68
24	.jpg	150975	33666	22,30	29668	19,65	25397	16,82
25	.jpg	4320000	357730	8,28	323642	7,49	348949	8,08
26	.jpg	405000	19888	4,91	17036	4,21	24338	6,01
27	.jpg	150876	8928	5,92	7440	4,93	8928	5,92
28	.jpg	151320	20698	13,68	18236	12,05	17614	11,64
29	.jpg	151200	36274	23,99	31182	20,62	27073	17,91
30	.jpg	692640	88806	12,82	73570	10,62	78407	11,32
31	.jpg	2508462	462388	18,43	417756	16,65	349480	13,93
32	.jpg	2477466	399110	16,11	361008	14,57	310751	12,54
33	.jpg	2477466	414148	16,72	376172	15,18	319813	12,91
34	.jpg	4536000	810856	17,88	726392	16,01	618924	13,64
35	.jpg	1083456	179226	16,54	160424	14,81	140336	12,95
36	.jpg	1193157	143086	11,99	127782	10,71	121015	10,14
37	.jpg	456750	101156	22,15	87030	19,05	75887	16,61
38	.jpg	1122000	128608	11,46	114002	10,16	110860	9,88
39	.jpg	165240	37514	22,70	32232	19,51	27817	16,83
40	.jpg	1570020	285532	18,19	256846	16,36	214579	13,67
Gamtos paveikslėliai								
41	.jpg	4800000	798432	16,63	711438	14,82	630657	13,14
42	.jpg	2359296	326818	13,85	293624	12,45	266531	11,30
43	.jpg	3888000	708590	18,23	620900	15,97	555968	14,30
44	.jpg	1200000	113234	9,44	105244	8,77	101394	8,45
45	.jpg	3932160	801130	20,37	702852	17,87	608854	15,48
46	.jpg	5760000	968464	16,81	837636	14,54	778331	13,51
47	.jpg	3147264	484554	15,40	418414	13,29	402720	12,80
48	.jpg	4800000	1241098	25,86	1083256	22,57	905288	18,86
49	.jpg	2088960	330746	15,83	307510	14,72	265588	12,71
50	.jpg	2359296	368882	15,64	321870	13,64	300803	12,75
51	.jpg	1157238	176452	15,25	148384	12,82	143577	12,41
52	.jpg	2073600	571004	27,54	497564	24,00	417352	20,13
53	.jpg	718200	90926	12,66	82900	11,54	74675	10,40
54	.jpg	1005600	152652	15,18	132116	13,14	122498	12,18
55	.jpg	1966080	288564	14,68	252756	12,86	229741	11,69
56	.jpg	790020	193870	24,54	168292	21,30	143869	18,21
57	.jpg	2359296	441952	18,73	389652	16,52	339022	14,37
58	.jpg	562500	131734	23,42	112388	19,98	100117	17,80
59	.jpg	6912000	467420	6,76	424580	6,14	463899	6,71
60	.jpg	1822500	284350	15,60	250142	13,73	227011	12,46
Grafikų ir diagramų paveikslėliai								
61	.jpg	867825	140556	16,20	120400	13,87	113092	13,03
62	.jpg	4962045	351378	7,08	310394	6,26	361621	7,29
63	.jpg	944640	73098	7,74	67268	7,12	72883	7,72
64	.jpg	988812	72142	7,30	67328	6,81	72210	7,30
65	.jpg	780669	122880	15,74	101564	13,01	99485	12,74
66	.jpg	430080	74774	17,39	63884	14,85	60501	14,07
67	.jpg	450858	55572	12,33	47392	10,51	47724	10,59
68	.jpg	469404	115348	24,57	94682	20,17	84627	18,03
69	.jpg	721800	105186	14,57	87842	12,17	89026	12,33
70	.jpg	524724	76532	14,59	66866	12,74	62445	11,90

71 .jpg	499500	66410	13,30	57358	11,48	57285	11,47
72 .jpg	716943	48414	6,75	44984	6,27	49799	6,95
73 .jpg	275520	40378	14,66	35156	12,76	33441	12,14
74 .jpg	443625	40652	9,16	37320	8,41	37604	8,48
75 .jpg	1531200	179716	11,74	159616	10,42	157498	10,29
76 .jpg	374850	39950	10,66	35514	9,47	35534	9,48
77 .jpg	243000	57386	23,62	48766	20,07	43251	17,80
78 .jpg	227853	43392	19,04	38384	16,85	32940	14,46
79 .jpg	442260	82740	18,71	67036	15,16	66832	15,11
80 .jpg	282240	51960	18,41	45300	16,05	40793	14,45
Miestų paveikslėliai							
81 .jpg	493050	74776	15,17	64112	13,00	59129	11,99
82 .jpg	134250	36994	27,56	31290	23,31	26647	19,85
83 .jpg	773760	227616	29,42	206152	26,64	170280	22,01
84 .jpg	699033	151198	21,63	129518	18,53	112455	16,09
85 .jpg	562500	162892	28,96	141048	25,08	115974	20,62
86 .jpg	1440000	327108	22,72	282514	19,62	244116	16,95
87 .jpg	968100	217042	22,42	190354	19,66	165679	17,11
88 .jpg	608652	177086	29,09	155598	25,56	125874	20,68
89 .jpg	1166400	265452	22,76	220208	18,88	207723	17,81
90 .jpg	583200	118374	20,30	101036	17,32	87576	15,02
91 .jpg	633360	88158	13,92	75846	11,98	72001	11,37
92 .jpg	402300	124930	31,05	106652	26,51	90619	22,53
93 .jpg	568980	101004	17,75	87952	15,46	76824	13,50
94 .jpg	593616	123916	20,87	108860	18,34	91788	15,46
95 .jpg	562500	93576	16,64	82466	14,66	73701	13,10
96 .jpg	630000	125718	19,96	108112	17,16	98481	15,63
97 .jpg	218610	43162	19,74	37508	17,16	32707	14,96
98 .jpg	745500	128424	17,23	115800	15,53	102684	13,77
99 .jpg	204480	52060	25,46	44952	21,98	37630	18,40
100 .jpg	2098176	502154	23,93	439918	20,97	371894	17,72
Bendras suglaudintų duomenų apimties santykis su neglaudintų duomenų apimtimi, %			18,91	-	16,46	-	14,77
Bendras algoritmų suglaudintų duomenų apimties santykis su A&A algoritmo suglaudintų duomenų apimtimi, %			21,90	-	10,26	-	-

Naudojant 50% kvantavimo matricą (tai atitinka 50 proc. glaudinimo lygį arba tiek pat procentų kokybės), rezultatai parodė, kad visais atvejais A&A algoritmas buvo efektyvesnis² (14,77% neglaudintų duomenų apimties) ir už standartinį RLE (18,91% neglaudintų duomenų apimties) ir už jo JPEG modifikaciją (16,46% neglaudintų duomenų apimties). Vidutiniškai A&A metodu suglaudintos bylos užimdavo 10,26 % mažiau vietos nei koduojant JPEG skirta RLE modifikacija, ir net 21,90 % mažiau vietos, nei koduojant standartiniu RLE. Lentelėje 3.3 lent. pateikiama informacija, kaip kiekvienas iš algoritmų suglaudino analizuojamus paveikslėlius pagal jų tipą (tekstūriniai, dvispalviai ir t.t.), kokie yra geriausi, blogiausi ir vidutiniai glaudinimo rezultatai.

² Šis teiginys remiasi bendrais visų tipų paveikslėlių glaudinimo rezultatais.

3.3 lentelė Algoritmų tarpusavio palyginimas (kvantavimo matrica 50%)

	Paveikslėlių tipas	Tekstūrų	Dvispalvių	Knygų puslapių	Gamtos	Grafikų ir diagramų	Miestų	Geriausi glaudinimo rezultatai	Blogiausi glaudinimo rezultatai	Vidutinis glaudinimo rezultatas
Algoritmų suglaudintų duomenų apimties santykis su neglaudintų duomenų apimtimi, %										
Algoritmas	RLE	25,50	13,68	17,22	17,12	14,18	22,33	13,68	25,50	18,91
	RLE modifikacija	22,07	11,92	15,30	15,03	12,22	19,37	11,92	22,07	16,46
	A&A	19,49	11,45	13,31	13,48	11,78	16,73	11,45	19,49	14,77
Algoritmų suglaudintų duomenų apimties santykis su A&A algoritmo suglaudintų duomenų apimtimi, %										
Algoritmas	RLE	23,57	16,27	22,68	21,25	16,90	25,08	16,27	25,08	21,90
	RLE modifikacija	11,68	3,96	13,00	10,32	3,62	13,62	3,62	13,62	10,26

Lentelėje 3.3 lent. geriausi rezultatai išskirti mėlsva, o blogiausi raudona spalva. Kaip matyti iš šių rezultatų, naudojant 50% glaudinimo lygį, nepasitvirtino prielaida, kad tekstūrų paveikslėlių kvantuotų matricių elementų sekas RLE algoritmai glaudins gerai. Rezultatai čia prasčiausi visų trijų algoritmų, nors A&A algoritmas buvo efektyvesnis (19,49% neglaudintų duomenų apimties), nei kiti du (25,50% RLE ir 22,07% RLE modifikacijos). Geriausius rezultatus RLE ir A&A algoritmai parodė glaudinant dvispalvių paveikslėlių duomenis. Nors JPEG algoritmas dvispalvių paveikslėlių vaizdus glaudina blogai (žr. 1.7.2 skyrių), tačiau kaip minėta 1.11 skyriuje, gautose sekose yra daug nuosekliai pasikartojančių vienarūšių duomenų. RLE tipo algoritmai čia veikia labai efektyviai, jų efektyvumas nežymiai skyrėsi ir nuo A&A algoritmo, kuris parodė geriausią suspaudimo efektą- koduoti duomenys sudarė tik 11,45% neglaudintų duomenų apimties. Miestų paveikslėlių duomenis, A&A algoritmas glaudino efektyviau (16,73% neglaudintų duomenų apimties), nei kiti du algoritmai. Lyginant algoritmus tarpusavyje- tai prasčiausias abiejų RLE algoritmų rezultatas (RLE 22,33% ir RLE modifikacijos 19,37%).

3.5.2. Antros dalies bandymo rezultatai

Antroje dalyje vaizdams glaudinti naudojama JPEG standarto 70% kvantavimo matrica (glaudinimas 30 %). Sekos koduojamos abejais RLE bei A&A algoritmais. Algoritmų tarpusavio palyginimo galutiniai rezultatai pateikti lentelėje 3.4 lent., o visi programinės įrangos išvesties Rezultatai pateikti 6.3 priedas. Antros dalies bandymų rezultatai

Naudojant 70% kvantavimo matricą, eksperimento metu gauti rezultatai parodė, kad visais atvejais A&A algoritmas buvo efektyvesnis³ už RLE algoritmus. Vidutiniškai A&A metodu suglaudintos bylos užimdavo 15,32 % mažiau vietos nei koduojant JPEG skirta RLE modifikacija, ir 29,34 % mažiau vietos, nei koduojant standartiniu RLE. Lentelėje 3.4 lent. pateikiama informacija, kaip kiekvienas iš algoritmų suglaudino analizuojamus paveikslėlius pagal jų tipą.

³ Šis teiginys remiasi bendrais visų tipų paveikslėlių glaudinimo rezultatais.

3.4 lentelė Algoritmų tarpusavio palyginimas (kvantavimo matrica 70%)

Paveikslėlių tipas		Tekstūrų	Dvispalvių	Knygų puslapių	Gamtos	Grafikų ir diagramų	Miestų	Geriausi glaudinimo rezultatai	Blogiausi glaudinimo rezultatai	Vidutinis glaudinimo rezultatas
Algoritmų suglaudintų duomenų apimties santykis su neglaudintų duomenų apimtimi, %										
Algoritmas	RLE	28,04	17,22	21,37	22,85	17,19	31,07	17,19	31,07	23,69
	RLE modifikacija	23,68	14,70	19,26	20,10	15,00	27,40	14,70	27,40	20,63
	A&A	20,61	13,64	15,43	16,82	13,44	21,95	13,44	21,95	17,47
Algoritmų suglaudintų duomenų apimties santykis su A&A algoritmo suglaudintų duomenų apimtimi, %										
Algoritmas	RLE	26,50	20,82	27,78	26,37	21,80	29,34	20,82	29,34	26,24
	RLE modifikacija	12,98	7,23	19,88	16,30	10,38	19,89	7,23	19,89	15,32

Lentelėje 3.4 lent. geriausi rezultatai išskirti mėlyna, o blogiausi raudona spalva. Naudojant 70% kvantavimo matricą, susumavus rezultatus, matyti, jog šiuo atveju visi algoritmai prasčiausiai glaudino miestų paveikslėlių kvantuotų matricų elementus. Atsižvelgus į algoritmų suglaudintų duomenų apimties santykį su A&A algoritmo suglaudintų duomenų apimtimi- RLE (29,34%) ir RLE modifikacija (19,89%), galima pastebėti, kad glaudindamas miestų paveikslėlių duomenų sekas, A&A algoritmas buvo efektyvesnis pirmojo (25,08% RLE ir 13,62% RLE modifikacija), o ne antro bandymo metu. Taip yra todėl, kad renkantis žemą suspaudimo lygį, JPEG algoritmas kvantavimo metu pašalina mažiau aukštadažnių spalvų dedamųjų- kvantuotose matricose dauguma elementų yra ne nuliai, dažniausiai skirtingi skaičiai, išsidėstę galimai mažėjimo tvarka. Tokios sekos RLE algoritmų koduojamos neefektyviai (kiekvienas pavienis elementas sudedamas į dviejų baitų paketus), o A&A algoritmas koduoja tik tas vietas kur yra pasikartojimai. Tai lemia, jog tokiose sekose išryškėja A&A algoritmo universalumas, nors efektyvumas yra mažesnis.

Geriausius rezultatus modifikuotas RLE algoritmas parodė glaudinant dvispalvių paveikslėlių duomenis (14,70% neglaudintų duomenų apimties), o standartinis RLE grafikų ir diagramų (17,19% neglaudintų duomenų apimties). A&A algoritmas parodė panašius rezultatus dvispalvių (13,64%) bei grafikų ir diagramų (13,44%) paveikslėlių glaudinime, nors pastarųjų sekas kodavo šiek tiek efektyviau. Iš šių rezultatų galima spręsti, jog A&A efektyvumas, kaip ir RLE algoritmų, priklauso nuo koduojamų duomenų tipo, tačiau yra mažiau jautrus ilgų vienodų simbolių sekų nebuvimui.

3.5.3. Trečios dalies bandymo rezultatai

Trečioje dalyje paveikslėliukams glaudinti naudojama JPEG standarto 30% kvantavimo matrica (glaudinimas 70 %). Sekos koduojamos abejais tiek RLE, tik A&A algoritmais. Algoritmų tarpusavio palyginimo galutiniai rezultatai pateikti lentelėje- 3.5 lent. , o visi programinės įrangos išvesties rezultatai pateikiami priede 6.4 priedas. Trečios dalies bandymų rezultatai

3.5 lentelė Algoritmų tarpusavio palyginimas (kvantavimo matrica 30%)

Paveikslėlių tipas	Tekstūrų	Dvispalvių	Knygų puslapių	Gamtos	Grafikų ir diagramų	Miestų	Geriausi glaudinimo rezultatai	Blogiausi glaudinimo rezultatai	Vidutinis glaudinimo rezultatas	
Algoritmų suglaudintų duomenų apimties santykis su neglaudintų duomenų apimtimi, %										
Algoritmas	RLE	14,88	11,07	14,22	12,53	11,99	16,86	11,07	16,86	13,78
	RLE modifikacija	12,37	9,77	12,60	11,09	10,31	14,63	9,77	14,63	11,92
	A&A	12,41	9,77	11,59	10,57	10,48	13,34	9,77	13,34	11,49
Algoritmų suglaudintų duomenų apimties santykis su A&A algoritmo suglaudintų duomenų apimtimi, %										
Algoritmas	RLE	16,63	11,70	18,51	15,68	12,56	20,89	11,70	20,89	16,59
	RLE modifikacija	-0,30	0,03	7,99	4,70	-1,61	8,79	-1,61	8,79	3,54

Lentelėje 3.5 lent. geriausi rezultatai išskirti mėlsva, o blogiausi raudona spalva. Naudojant 30% kvantavimo matricą, iš pateiktų rezultatų matyti, jog visi algoritmai, kaip ir antro bandymo metu, neefektyviausiai glaudino miestų paveikslėlių duomenų sekas. Čia verta pastebėti, jog didžiausias (29,34% geriau nei RLE ir 19,89% geriau nei RLE modifikacija) A&A algoritmo pranašumas buvo antrojo bandymo metu, koduojant miestų paveikslėlius. Šiuos rezultatus plačiau aptarsime sekančiame skyriuje. Atlikus trečiąjį bandymą ir atsižvelgus į pirmojo bandymo rezultatus, paaiškėjo, jog esant didesniems suspaudimo laipsniams (naudojant 50 ir 30 proc. kvantavimo matricas), visi algoritmai geriausiai glaudina dvispalvius paveikslėlius, nes juose yra pakankamai ilgos pasikartojančių vienodų simbolių sekos. Jei glaudinimo laipsnis aukštas, tai pastarosios sekos dar labiau pailgėja, nes vis daugiau kvantuotų matricų elementų yra skaičius „0“. Trečiojo bandymo metu taip pat išryškėjo ir RLE modifikacijos pranašumas, glaudinant tekstūrų bei grafikų ir diagramų paveikslėlius. Geresnį tokių paveikslėlių glaudinimą, naudojant modifikuotą RLE, galima būtų sieti su JPEG būdingu spalvų koreliacijos mažinimu. Kadangi tokio tipo paveikslėliuose spalvų yra nedaug ir jos skiriasi nežymiai, taikant didesnę glaudinimo lygį, JPEG algoritmas po kvantavimo, palieka tik 8x8 bloko DC koeficientą, visus AC koeficientus prilyginant nuliui. Kaip žinome, iš tokio bloko elementų sudarytos sekos, koduojamos efektyviai, o ypač modifikuoto RLE algoritmo.

3.5.4. Apibendrinti bandymų rezultatai

Iš 3.6 lent.3.6 lentelė pateiktų rezultatų matyti, koks yra vidutinis kiekvieno algoritmo duomenų suglaudavimo efektyvumas trijų bandymų metu bei šių efektyvumo rodiklių palyginimas su A&A algoritmu. Vidurkis apskaičiuojamas pagal (22) formulę

$$\mu = \frac{1}{N-1} \sum_{i=0}^{N-1} x_i \quad (22)$$

čia N – reikšmių skaičius, i – reikšmės numeris.

Vertinat bendrai, A&A algoritmu užkoduoti duomenys vidutiniškai užima net 22,43% mažiau vietos nei RLE, ir 10,76% mažiau nei RLE JPEG modifikacija, o jo vidutinis glaudinimo efektyvumas (neglaudintų ir suglaudintų duomenų apimties santykis) yra mažiausias-14,58% (3.6 lent.).

3.6 lentelė Vidutinio algoritmų efektyvumo palyginimas su A&A algoritmu efektyvumu

Algoritmas	Vidutinis glaudinimo efektyvumas ⁴ , %	Algoritmų efektyvumo santykis su A&A algoritmo efektyvumu, %
RLE	18,79	22,43
RLE modifikacija	16,34	10,76
A&A	14,58	

3.7 lent. pateikti blogiausi ir geriausi algoritmų rezultatai, pagal paveikslėlių tipus, naudojant tirs skirtingus glaudinimo lygius. Rezultatas skaičiuojamas imant nesuglaudintų duomenų ir suglaudintų duomenų santykį procentais.

3.7 lentelė Blogiausių ir geriausių algoritmų glaudinimo rezultatų suvestinė

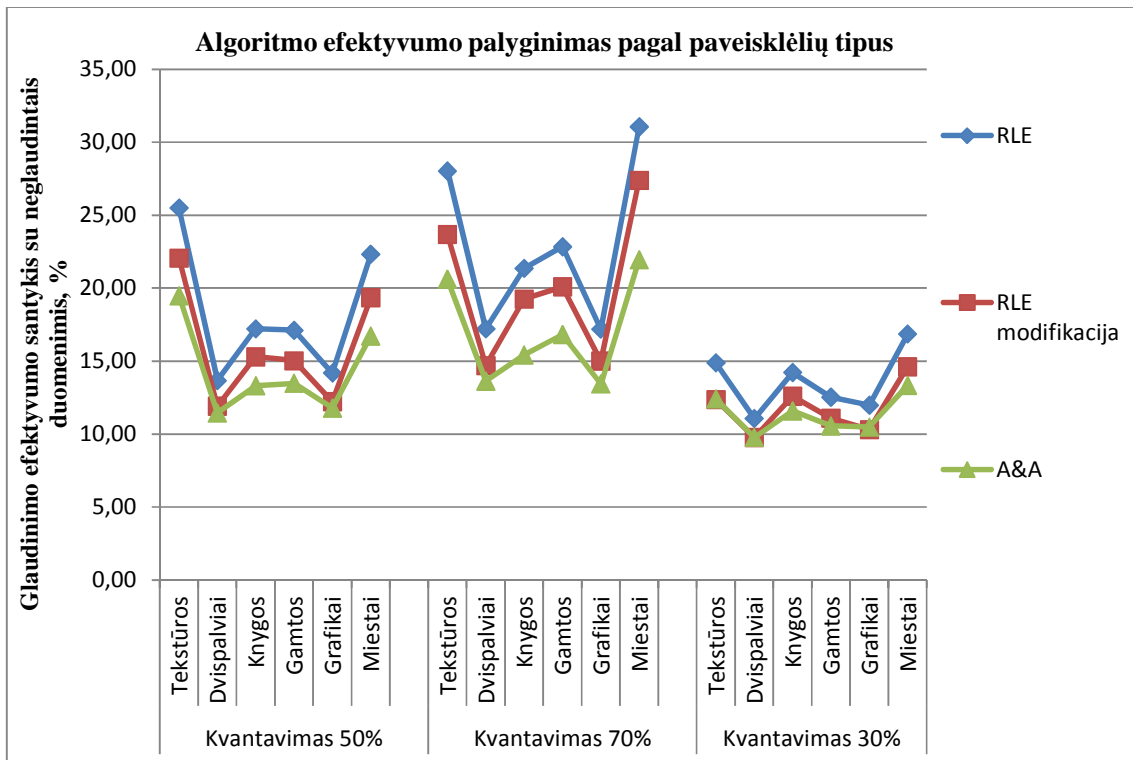
Kvantavimas, %	Blogiausi rezultatai		Geriausi rezultatai	
	Tipas	Rezultatas, %	Tipas	Rezultatas, %
RLE				
50%	Tekstūros	25,50	Dvispalviai	13,68
70%	Miestai	31,07	Grafikai ir diagramos	17,19
30%	Miestai	16,86	Dvispalvių	11,07
RLE modifikacija				
50%	Tekstūrų	22,07	Dvispalvių	11,92
70%	Miestų	27,40	Dvispalvių	14,70
30%	Miestų	14,63	Dvispalvių	9,77
A&A				
50%	Tekstūrų	19,49	Dvispalvių	11,45
70%	Miestų	21,95	Grafikų ir diagramų	13,44
30%	Miestų	13,34	Dvispalvių	9,77

Iš 3.7 lent. duomenų matyti, jog blogiausius glaudinimo rezultatus visi algoritmai pasiekė koduojant tekstūrų paveikslėlius, kai kvantavimas 50% ir miestų, kai kvantavimas 30 ir 70%.

Pirmu atveju suspaudimo lygis yra vidutinis, taigi kvantavimo žingsnyje dar paliekama pakankamai daug aukštadažnių spalvų dedamųjų, elementai nesuapvalinami iki nulio, o tai lemia, jog nesusidaro ilgos sekos- krenta kodavimo efektyvumas. Miestų paveiksliai pasižymi žymiais kontrastais, o vienalyčių tekstūrų atsikartojimų nepakanka, kad kodavimas būtų efektyvus. Iš rezultatų matyti, jog ilgų sekų nebuvimo miestų paveikslėlių duomenyse problema, įtakoja mažą kodavimo efektyvumą, naudojant 30 ir 70% kvantavimą. Nors rezultatai ir blogiausi, visgi A&A algoritmas buvo pats efektyviausias, palyginus su RLE, o didžiausias pranašumas pasiektas glaudinant miestų paveikslėlius, esant mažam suspaudimo laipsniui.

Peržiūrint geriausius rezultatus, galima teigti, jog esant skirtingiems glaudinimo lygiams, visi algoritmai tinkamiausi glaudinti dvispalvius ir grafikų bei diagramų paveikslėlius. Pažymėtina, jog naudojant didelį suspaudimo laipsnį (kvantavimas 30%), A&A algoritmo ir JPEG RLE modifikacijos rezultatas glaudinant dvispalvius paveikslėlius buvo vienodas. Taip yra dėl pastarojo algoritmo savybės ypatingai gerai koduoti „0“ sekas.

⁴ Neglaudintų ir suglaudintų duomenų apimties santykis procentais



3.5 pav. Algoritmo efektyvumo palyginimas pagal paveikslėlių tipus

3.5 pav. grafiškai pavaizduotas trijų algoritmų efektyvumas pagal paveikslėlių tipus, esant skirtingiems glaudinimo lygiams. Efektyvumas skaičiuojamas imant suglaudintų ir nesuglaudintų duomenų santykį. Šiame paveikslėlyje matome ir 3.7 lent. pateiktus duomenis- blogiausius ir geriausius algoritmų rezultatus.

Panagrinėkime kokių procentiniu skirtumu nuo vidutinio suglaudavimo efektyvumo pasiskirstė kodavimo efektyvumas pagal paveikslėlių tipus. Palyginimo rezultatai pateikiami 3.8 lent.

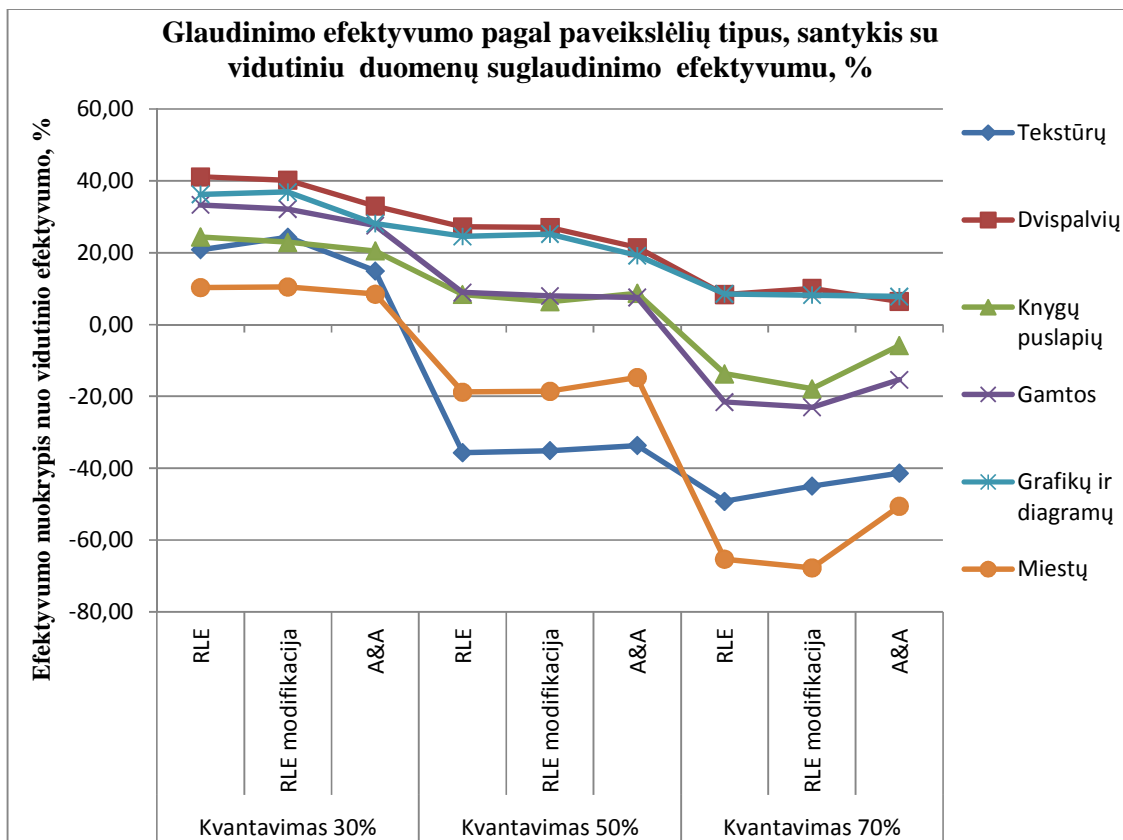
3.8 lentelė Glaudinimo efektyvumas pagal paveikslėlių tipus, lyginant su vidutiniu duomenų suglaudinimo efektyvumu.

Paveikslėlių tipas	Tekstūrų	Dvispalvių	Knygų puslapių	Gamtos	Grafikų ir diagramų	Miestų	Algoritmo efektyvumo vidutinis nuokrypis nuo bendro vidutinio glaudinimo efektyvumo, %	Bendras algoritmų efektyvumo vidutinis nuokrypis nuo bendro vidutinio glaudinimo efektyvumo, %	Mažiausias algoritmų glaudinimo efektyvumo vidutinis nuokrypis nuo bendro vidutinio glaudinimo efektyvumo, %	
										Glaudinimo efektyvumo pagal paveikslėlių tipus, santykis su vidutiniu duomenų suglaudinimo efektyvumu, %
Algoritmas	Kvantavimas 50 % (glaudinimo lygis 50%)							1,99	1,41	
	RLE	-35,68	27,21	8,37	8,91	24,55	-18,82			2,42
	RLE modifikacija	-35,10	27,03	6,34	7,99	25,19	-18,58			2,15
	A&A	-33,69	21,46	8,70	7,53	19,20	-14,76	1,41		
	Kvantavimas 70 % (glaudinimo lygis 30%)							-20,40	-16,49	
	RLE	-49,20	8,37	-13,71	-21,58	8,53	-65,32			-22,15
	RLE modifikacija	-44,96	10,01	-17,90	-23,04	8,18	-67,73			-22,57
	A&A	-41,37	6,44	-5,84	-15,38	7,81	-50,57	-16,49		
	Kvantavimas 30 % (glaudinimo lygis 70%)							25,85	22,08	
	RLE	20,83	41,12	24,33	33,32	36,22	10,27			27,68
	RLE modifikacija	24,29	40,17	22,89	32,12	36,86	10,46			27,80
	A&A	14,91	32,98	20,50	27,51	28,11	8,48	22,08		

3.8 lent. pateikti duomenys suskaičiuoti pagal (23) formulę, palyginant bendrą vidutinį efektyvumą (suglaudintų duomenų apimtį santykiu su nesuglaudintų duomenų apimtimi procentais) su kiekvieno algoritmo efektyvumu, koduojant atskirus paveikslėlių tipus, esant skirtingiems glaudinimo laipsniams.

$$d = \frac{1}{N-1} \sum_{i=0}^{N-1} (\mu - x_i) \quad (23)$$

Vidutinis nuokrypis pasako vidutinį intervalo plotį, kuriame išsidėsčiusios šaltinio reikšmės. Tokiu būdu gaunamas procentais išreikštas vidutinis nuokrypis nuo vidutinio glaudinimo efektyvumo. Lentelėje 3.8 lent. taip pat pateikti duomenys, koks buvo vidutinis nuokrypis (%), esant skirtingiems glaudinimo laipsniams, nuo bendro vidutinio glaudinimo efektyvumo. Šis paskaičiavimas atliktas kiekvienam iš algoritmų atskirai, visiems algoritmams bendrai bei išskirtas mažiausias nuokrypis.



3.6 pav. Glaudinimo efektyvumo pagal paveikslėlių tipus palyginimas su vidutiniu duomenų suglaudinimo efektyvumu

3.6 pav. grafiškai pavaizduotas glaudinimo efektyvumo pagal paveikslėlių tipus palyginimas su vidutiniu duomenų suglaudinimo efektyvumu. Iš pastarojo paveikslėlio ir iš 3.8 lent. pateiktų rezultatų matyti, kad imant bendrai visu algoritmus, didžiausi nuokrypiai užfiksuoti esant 30 ir 70 % glaudinimo laipsniams, o paveikslėlius glaudinant 50%, nuokrypis buvo tik 1,99%. Iš to seka išvada, jog norint ir efektyviai suglaudinti paveikslėlių vaizdus JPEG algoritmu, išlaikyti gerą vaizdo kokybę ir efektyviai koduoti susidariusias sekas analizuojamais algoritmais, patartina naudoti standartinę 50% kvantavimo matricą.

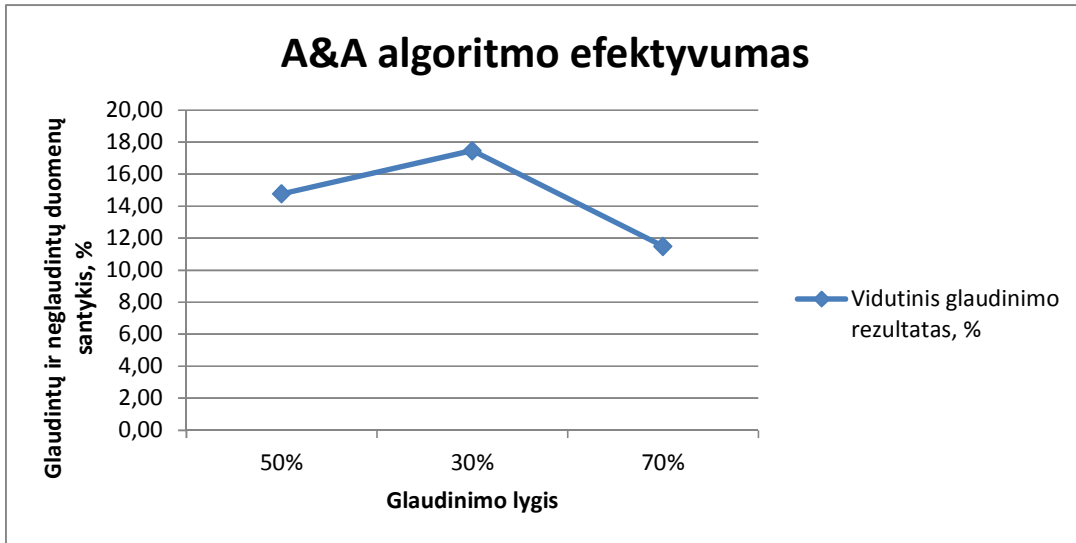
Didžiausias teigiamas visų algoritmų rezultatas (25,85%), pasiektas naudojant aukštą (70%) glaudinimo lygį. Taigi ir algoritmų efektyvumas čia didžiausias. Jei suglaudinto vaizdo kokybei nekeliama aukšti reikalavimai, galima žymiai sumažinti vaizdo bylos apimtį, nes ir visi analizuojami entropiniai algoritmai taip suglaudinto vaizdo sekas koduoja ypač efektyviai.

Glaudinimui naudojant JPEG standartą, kvantavimui 70 % matricą t.y. glaudinimo lygis-30%, pasiekiami blogiausi rezultatai, nes nuokrypis nuo vidutinio efektyvumo yra neigiamas ir lygus -20,40%. A&A algoritmo vidutinis efektyvumo nuokrypis pagal paveikslėlių tipus, nuo jo bendro vidutinio efektyvumo yra -16,49%, atitinkamai RLE yra -22,15% ir RLE modifikacijos yra -22,57%. Įvertinus skirtumą tarp algoritmų efektyvumo vidutinio nuokrypio nuo bendro vidutinio glaudinimo efektyvumo, didžiausi skirtumai gaunami naudojant mažą (30%) glaudinimo lygį. Skirtumų apskaičiavimas pateiktas žemiau.

- Tarp RLE ir A&A algoritmo: $(|-22,15\%| - |-16,49\%| = |-5,66\%| > (|27,68\%| - |22,80\%| = |5,60| > (|2,42\%| - |1,41\%| = |1,02|))$;
- Tarp RLE modifikacijos ir A&A algoritmo: $(|-22,57\%| - |-16,49\%| = |-6,08\%| > (|27,80\%| - |22,08\%| = |5,72| > (|2,15\%| - |1,41\%| = |0,74|))$;

Kaip matome iš 3.8 lent. ir šių skaičiavimo rezultatų, didžiausias A&A algoritmo efektyvumo skirtumas, lyginant su kitais dviem, fiksuojamas esant mažam glaudinimo lygiui. Tai patvirtina prielaidą, kad A&A algoritmas bus efektyviausias koduojant mažai glaudintų vaizdų sekoms koduoti, o jo savybė mažiausiai nukrypti nuo vidutinio glaudinimo efektyvumo, suteikia

pagrindo teigti, kad jis yra universalus visų tipų vaizdo duomenų sekoms koduoti. Prie tokių tipų galima priskirti tiek miestų tiek ir gamtos vaizdus, kurie pasižymi ryškiais kontrastais ir spalvų įvairove.



3.7 pav. A&A algoritmo kodavimo efektyvumas, esant skirtingiems glaudinimo laipsniams

A&A algoritmo kodavimo efektyvumo priklausomybė nuo suglaudavimo lygio grafiškai pavaizduota 3.7 pav. Šis rezultatas byloja, kad geriausias duomenų suglaudimas A&A algoritmu, pasiekiamas esant aukštam suglaudavimo lygiui.

4. IŠVADOS

1. Atlikti tyrimai įrodė, jog šiuo standartu glaudintų vaizdų kvantuotų matricių elementų sekas, geriausiai suglaudina A&A algoritmas, nes jo vidutinis suglaudavimo rezultatas (žr. 3.6 lent.) yra 14,58% neglaudintų duomenų apimties, kai RLE algoritmo- 18,79%, o modifikuoto RLE algoritmo- 16,34%.
2. Atliekant bandymus, A&A algoritmo naudingumas palygintas su RLE ir JPEG standarte naudojama RLE modifikacija. Remiantis 3.3 lent., 3.4 lent. ir 3.5 lent. pateiktais vidutiniais glaudinimo rezultatais, galima teigti, jog naudojant bet kokio lygio glaudinimą ir bet kokius paveikslėlių tipus, bendras A&A algoritmo efektyvumas buvo aukščiausias- suglaudinti duomenys užimdavo mažiausiai vietas.
3. Realizuota programinė įranga, visiškai atitinkanti JPEG standarto reikalavimus, kurioje autos sekos yra koduojamos integruotu A&A algoritmu. Toks JPEG pakeitimas (papildymas) užtikrina efektyvesnę tiek dvispalvių, grafikų bei diagramų, tiek ir didesnės spalvinės raiškos vaizdų kodavimą, esant mažesniai vaizdo glaudinimo laipsniui
4. Naudojant vaizdo ir jo fragmentų duomenų bazę, eksperimentiškai patikrintas sudaryto simbolių sekos ilgio kodavimo algoritmo A&A naudingumas. Išanalizavus gautus rezultatus, išskirti sudarytojo algoritmo privalumai ir trūkumai.
 - 4.1. A&A algoritmo privalumai:
 - 4.1.1. Dėl savo specifikos A&A algoritmas, kitaip ne išbandytieji RLE, niekada nepadidins koduojamų duomenų apimties.
 - 4.1.2. Paprastesnis veikimas- nereikia naudoti baitų paketų.
 - 4.1.3. Universalus panaudojimo atžvilgiu, kadangi esant skirtingiems glaudinimo lygiams ir paveikslukams, parodė mažiausią nuokrypį nuo savo vidutinio glaudinimo efektyvumo.
 - 4.1.4. Didžiausią pranašumą (lyginant su RLE) pasiekia esant žemam glaudinimo lygiui. Tai įrodo, kad A&A algoritmas gali efektyviai koduoti ir mažai glaudintų paveikslėlių sekas.
 - 4.1.5. Efektyviausias dvispalvių ir grafikų bei diagramų paveikslukų duomenų sekoms koduoti. Gerai koduoja didelės spalvinės raiškos (gamtos) paveikslukų duomenų sekas.
 - 4.2. A&A algoritmo trūkumai:
 - 4.2.1. Pasižymi išimtimi, kuomet A&A algoritmas praranda savo pranašumą- jei visą seką sudaro tas pats simbolis. RLE pakanka 2 baitų, A&A sunaudoja papildomą baitą įterpdamas žymeklį.
 - 4.2.2. Mažiausias efektyviai užkoduotas sekos ilgis yra keturi elementai, kai RLE tik trys.
5. Tolimesni darbo tikslai. Šiame darbe teiginys, jog RLE tipo algoritmai veikia paprastai ir greitai, yra paremtas literatūros šaltinyje [12] minima informacija bei plačiu šių algoritmų panaudojimu įvairiose srityse. Bandymų metu nebuvo atsižvelgiama į RLE ir A&A algoritmų greitaveiką. Norint pilnavertiškai įvertinti ir palyginti analizuotus algoritmus, jų veikimo greitis turėtų būti įvertintas skaitiškai. Tolimesni darbo tikslai yra šie:
 - 5.1.1. A&A algoritmo greitaveikos analizė, jos palyginimas su RLE algoritmų greitaveika.
 - 5.1.2. A&A algoritmo veikimo patikimumo patikra- ar visada užkoduoti duomenys tinkamai dekoduojami.
 - 5.1.3. Ištirti ar efektyviai, lyginant su RLE, A&A algoritmu užkoduotos sekos, gali būti užkoduotos kitais entropiniais algoritmais (pvz. Hafmano). Patikrinti koks būtų bylos dydis, sudėjus vaizdo ir metaduomenis į galutinę bylos struktūrą.
 - 5.1.4. A&A algoritmo efektyvumo padidinimui, būtų galima realizuoti schemą, kuri elementų sekos ilgio skaitliukui naudotų tik 6 bitus ($2^6 = 64$), nes JPEG kiekviena 8x8 matrica koduojama atskirai, o įvertinus tai, maksimalus sekos ilgis yra 64 elementai. Taip būtų sutaupoma po 2 bitus, kiekvieną kartą kai A&A algoritmas koduoja ilgesnę nei 4 iš eilės pasikartojančių vienodų elementų seką.

5. LITERATŪRA

- [1] Melninkaitis, A. Vilniaus Universitetas, Lazerinių tyrimų centro seminaras „Skaitmeniniai vaizdai: registravimas, taikymai, skaitmeninė analizė“. [interaktyvus] [žiūrėta 2012-11-18]. Prieiga per internetą: <http://projektai.vu.lt/s-lasercenter/ppt/Andrius%20Melninkaitis%20LTC%20Seminaras%202006%2002%2023.pdf>
- [2] Jankauskas, K. Kauno technologijos universitetas, modulio P170M104 6 paskaita „JPEG vaizdų kodavimas“. [interaktyvus] [žiūrėta 2012-11-29]. Prieiga per internetą: <http://www.multimedija.ktu.lt/pages/moduliai/p170m104/paskaitos.html>.
- [3] Jankauskas, K. Kauno technologijos universitetas, modulio P170M104 5 paskaita „Statinio dvimačio vaizdo kodavimo standartai“. [interaktyvus] [žiūrėta 2012-11-29]. Prieiga per internetą: <http://www.multimedija.ktu.lt/pages/moduliai/p170m104/paskaitos.html>.
- [4] International Telecommunication Union, The international telegraph and telephone consultative committee, rekomendacija T.81 „Information technology- digital compression and coding of continuous- tone still images- Requirements and guidelines“. [interaktyvus] [žiūrėta 2013-01-15]. Prieiga per International Telecommunication Union svetainę: <http://www.itu.int/rec/T-REC-T.81/en>.
- [5] Digital Video and HD Algorithms and Interfaces, Second Edition. Sudarytojas Charles Poynton, 2012. 29 skyrius Component video color coding for SD. p.357.
- [6] Poynton, Ch. YUV and luminance considered harmful. [interaktyvus]. June 1999, [žiūrėta 2013-01-20], prieiga per internetą http://poynton.com/papers/YUV_and_luminance_harmful.html.
- [7] Poynton, Ch. The rehabilitation of gamma. [interaktyvus] [žiūrėta 2013-01-20], prieiga per internetą: http://www.poynton.com/PDFs/Rehabilitation_of_gamma.pdf.
- [8] Livingstone, M. The First Stages of Processing Color and Luminance: Where and What. *Vision and Art: The Biology of Seeing*. New York: Harry N. Abrams. 2002 pp. 46–67. ISBN 0-8109-0406-3.
- [9] Dusevičius, V. Efektyvaus vaizdų suspaudimo algoritmo sudarymas ir tyrimas. [interaktyvus]. Gegužė, 2004. [žiūrėta 2013-03-16]. Prieiga per KTU elektroninės bibliotekos sistemą.
- [10] Blelloch, G., E. Introduction to Data Compression. [interaktyvus]. January 2013; p. 5-29. [žiūrėta 2013-02-02]. Prieiga per Computer Science Department, Carnegie Mellon University prieigą: <http://blellochcs.cmu.edu>.
- [11] Cabeen, K.; Gent, P. Image Compression ant the Discrete Cosine Transform, College of the Redwoods. [interaktyvus] [žiūrėta 2013-02-10]. Prieiga per internetą: <http://msemac.redwoods.edu/~darnold/math45/laproj/Fall98/PKen/dct.rap>
- [12] The Data Compression Book, sudarytojas Nelson, M. R. Redwood City, CA, 1991.
- [13] Franti, P.; Survey, A.; Nevalainen, O.; Kaukoranta, T. Compression of Digital Images by Block Truncation Coding: A Survey. *The Computer Journal*, Vol. 37, No. 4, p. 308-332, 1994.
- [14] Mathematical Theory of Entropy. Sudarytojai: Martin, N. F.G.; James, E., W. 1984, p. 51. ISBN 978-0-521-17738-2.
- [15] Shannon, C. E. A Symbolic Analysis of Relay and Switching Circuits. *American Institute of Electrical Engineers*, vol. 57, No 12, p. 713–723.
- [16] Witten, I. H.; Neal, R. M.; Cleary, J. G. Arithmetic Coding for Data Compression. *Communications of the ACM*. Vol. 30 No. 6, p. 520–540, 1987.
- [17] Huffman, D.A. A Method for the Construction of Minimum-Redundancy Codes. *Proceedings of the I.R.E.*, p. 1098–1102, September 1952.
- [18] Smolinskas M. Dvejetainės informacijos kodavimo taikant bazinius skaidinius analizė: teoriniai ir praktiniai aspektai. 2005. [žiūrėta 2013-05-04]. Prieiga per KTU elektroninės bibliotekos sistemą
- [19] Ziv, J.; Lempel, A. Compression of Individual Sequences via Variable-Rate Coding". *IEEE Transactions on Information Theory*, Vol 24, No. 5, p. 530–536, September 1978.
- [20] Welch, T.A. A Technique for High-Performance Data Compression. *Computer*, vol. 17, No. 6, p. 8–19, June 1984.

6. PRIEDAI

6.1. priedas. Programinės įrangos kodas

```
function color()

    %-- Išvalomas ekranas, uždaromi langai. --%
    clc, close all, clear all;
    disp('-----');
    %-- Nurodoma kokius paveiksliukus nuskaityti --%
    for i = 101 : 101
        str = ['Eksperimentas Nr.', num2str(i)];
        disp(str);

        file = [num2str(i), '.jpg'];
        img = double(imread(file,'jpg'));
        img = img / 255;

        %-- Kokybės koeficientas (Q50). --%
        %-- Jeigu scale didinamas (> 1) kokybė gerėja. --%
        %-- Jeigu scale mažinamas (< 1) kokybė mažėja. --%
        %-- Testuojama buvo su scale=1 (kvantavimas 50%), scale=2 (kvantavimas 70%) ir
        scale=0.6 (kvantavimas 30%) . --%

        scale = 1;

        %-- Nuskaitomas aukštis ir plotis. --%
        width = size(img, 2);
        height = size(img, 1);
        r=[width * height * 3];

        %-- Apskaičiuojama kiek bus x ir y blokų. --%
        xblocks = width / 8;
        yblocks = height / 8;

        %-- Pagal kokybės koeficientą pakoreguojamos kvantavimo matricos. --%
        lumtable = lumquant(scale);
        chrtable = chrquant(scale);

        %-- Paveikslėlis iš RGB spalvų erdvės paverčiamas į YCbCr spalvų erdvę. --%
        img = RGB2YUV(img, width, height);

        %-- Kadangi matricoje reikšmės intervale [0; 1] dauginama iš 255. --%
        img = round(img * 255);

        %-- Kiekvienai matricai atliekamas konvertavimas į JPEG standartą.--%
        size1 = jpgtrans(img(:, :, 1), width, height, xblocks, yblocks, lumtable);
        size2 = jpgtrans(img(:, :, 2), width, height, xblocks, yblocks, chrtable);
        size3 = jpgtrans(img(:, :, 3), width, height, xblocks, yblocks, chrtable);

        size4(1) = size1(1) + size2(1) + size3(1);
        size4(2) = size1(2) + size2(2) + size3(2);
        size4(3) = size1(3) + size2(3) + size3(3);

        strJ = [num2str(size4(1))];
        strA = [num2str(size4(2))];
        strR = [num2str(size4(3))];

        Line1=['Neglaudinti paveikslėlio duomenys:      ',num2str(r), ' B'];
        disp(Line1);
        Line2=['RLE suglaudinti duomenys:              ',strR, ' B'];
        disp(Line2);
        Line3=['RLE modifikacija suglaudinti duomenys: ',strJ, ' B'];
        disp(Line3);
        Line4=['A&A suglaudinti duomenys:                ',strA, ' B'];
        disp(Line4);
    disp('-----');
end;

%-- Funkcija pakoreguoja kvantavimo matricą pagal kokybės koeficientą (scale), Y`
komponentė. --%
function qtable = lumquant(scale)

    qtable = [16, 11, 10, 16, 24, 40, 51, 61
              12, 12, 14, 19, 26, 58, 60, 55
              14, 13, 16, 24, 40, 57, 69, 56
              14, 17, 22, 29, 51, 87, 80, 62
```

```

18, 22, 37, 56, 68, 109, 103, 77
24, 35, 55, 64, 81, 104, 113, 92
49, 64, 78, 87, 103, 121, 120, 101
72, 92, 95, 98, 112, 100, 103, 99];

qtable = round(qtable ./ scale);
qtable(qtable < 1) = 1;
qtable(qtable > 255) = 255;

%--Funkcija pakoreguoja kvantavimo matricą pagal kokybės koeficientą (scale).C komponentės-%
function qtable = chrquant(scale)

qtable = [17, 18, 24, 47, 99, 99, 99, 99;
18, 21, 26, 66, 99, 99, 99, 99;
24, 26, 56, 99, 99, 99, 99, 99;
47, 66, 99, 99, 99, 99, 99, 99;
99, 99, 99, 99, 99, 99, 99, 99;
99, 99, 99, 99, 99, 99, 99, 99;
99, 99, 99, 99, 99, 99, 99, 99;
99, 99, 99, 99, 99, 99, 99, 99];

qtable = round(qtable ./ scale);
qtable(qtable < 1) = 1;
qtable(qtable > 255) = 255;

%-- Paveikslėlis iš RGB spalvų erdvės paverčiamas į YCbCr spalvų erdvę. --%
function x = RGB2YUV(x, width, height)
yuvmat = [ 0.299, 0.587, 0.114;
-0.1687, -0.3313, 0.5;
0.5, -0.4187, -0.0813];

for i = 1 : width
for j = 1 : height
x(j, i, :) = yuvmat * squeeze(x(j, i, :)) + [0; 0.5; 0.5];
end
end

%-- Paveikslėlis iš YCbCr spalvų erdvės paverčiamas į RGB formatą. --%
function x = YUV2RGB(x, width, height)
rgbmat = [1, 0, 1.402;
1, -0.34414, -0.71414;
1, 1.772, 0];

for i = 1 : width
for j = 1 : height
x(j, i, :) = rgbmat * (squeeze(x(j, i, :)) - [0; 0.5; 0.5]);
end
end

%-- Diskrečioji kosinuso transformacija (DCT). --%
function F = dct(f)

c = ones(1, 8) * sqrt(2 / 8);
c(1) = 1 * sqrt(1 / 8);

for u = 0 : 7
for v = 0 : 7
F(u + 1, v + 1) = c(u + 1) * c(v + 1) * sum(sum(f .* (cos((([0 : 7]' + 1 / 2)
* u * pi / 8) * cos((([0 : 7] + 1 / 2) * v * pi / 8)))));
end;
end;

%-- Atvirkštinė diskrečioji kosinuso transformacija (DCT). --%
function f = invdct(F)

c = ones(1, 8) * sqrt(2 / 8);
c(1) = 1 * sqrt(1 / 8);

for x = 0 : 7
for y = 0 : 7
f(x + 1, y + 1) = sum(sum(F .* (c' * c) .* (cos((x + 1 / 2) * [0 : 7]' * pi / 8)
* cos((y + 1 / 2) * [0 : 7] * pi / 8)))));
end;
end;

%-- DC ir AC koeficientų matricos nuskaitymas zigzago principu. --%
function x = zigzag(x)

zigzag = [ 1, 2, 6, 7, 15, 16, 28, 29
3, 5, 8, 14, 17, 27, 30, 43
4, 9, 13, 18, 26, 31, 42, 44
```

```

        10, 12, 19, 25, 32, 41, 45, 54
        11, 20, 24, 33, 40, 46, 53, 55
        21, 23, 34, 39, 47, 52, 56, 61
        22, 35, 38, 48, 51, 57, 60, 62
        36, 37, 49, 50, 58, 59, 63, 64];

zigzag = zigzag(:);
x = x(:);
x(zigzag) = x;

%-- DC ir AC koeficientų matricos nuskaitymas zigzago principu iš priešingos pusės. --%
function x = dezigzag(x)

    zigzag = [ 1,  2,  6,  7, 15, 16, 28, 29
              3,  5,  8, 14, 17, 27, 30, 43
              4,  9, 13, 18, 26, 31, 42, 44
              10, 12, 19, 25, 32, 41, 45, 54
              11, 20, 24, 33, 40, 46, 53, 55
              21, 23, 34, 39, 47, 52, 56, 61
              22, 35, 38, 48, 51, 57, 60, 62
              36, 37, 49, 50, 58, 59, 63, 64];

    zigzag = zigzag(:);
    x = reshape(x(zigzag), 8, 8);

%-- Skaičių sekos užkodavimas modifikuotu RLE algoritmu. --%
function output = RLE_mod(input)
    seq = 0;
    j = 1;
    l = length(input);

    for i = 1 : l
        if (input(i) == 0)
            seq = seq + 1;
        else
            output(j) = seq;
            output(j + 1) = input(i);
            j = j + 2;
            seq = 0;
        end;
    end;

    if (seq > 0)
        output(j) = 0;
        output(j + 1) = 0;
    end;

%-- Skaičių sekos užkodavimas A&A algoritmu. --%
function output = AA(input)
    j = 1;
    k = 1;
    m = 1;
    l = length(input);
    last = input(1);

    while (j < l-2)
        if (input(j) == input(j+1) & input(j) == input(j+2) && input(j) == input(j+3))
            seq = 4;
            output(k) = input(j);
            output(k+1) = 'A';
            m=j+4;
            while(m <= l && input(m) == input(j))
                seq = seq + 1;
                m=m+1;
            end;
            output(k+2) = seq;
            k=k+3;
            j=m;
        else
            output(k) = input(j);
            k=k+1;
            j=j+1;
        end;
    end;
    ff = l-j;
    for kint=0:ff
        output(k+kint)=input(j+kint);
    end;

%-- Skaičių sekos užkodavimas A&A algoritmu. --%
function output = RLE(input)

```



```

seq = 1;
j = 1;
l = length(input);
last = input(l);

for i = 2 : l
    if (input(i) == last)
        seq = seq + 1;
    else
        output(j) = last;
        output(j + 1) = seq;
        j = j + 2;
        seq = 1;
        last = input(i);
    end;
end;

output(j + 1) = last;
output(j) = seq;

%-- Paveiksliuko konvertavimas į JPEG standartą. --%
function size = jpgtrans(img, xsize, ysize, xblocks, yblocks, qtable)
    prevdc = 0;
    l1 = 0;
    l2 = 0;
    l3 = 0;

    %-- Einama per visus stulpelius ir eilutes (8x8 blokus). --%
    for j = 1 : xblocks
        for i = 1 : yblocks

            %-- Iškerpamas 8x8 blokas. --%
            imblock{i, j} = img((i - 1) * 8 + 1 : (i * 8), (j - 1) * 8 + 1 : (j * 8));

            %-- Reikšmės iš [0; 255] intervalo perslenkamos į [-128; 127].--%
            imblock{i, j} = imblock{i, j} - 128;

            %-- Atliekama diskrečioji kosinuso transformacija (DCT). --%
            dctcoef{i, j} = dct(imblock{i, j});

            %-- Kvantavimas. --%
            dctcoef{i, j} = round(dctcoef{i, j} ./ qtable);

            %-- Įsimenamas DC koeficientas. --%
            temp = dctcoef{i, j} (1, 1);

            %-- Išsaugomas tik DC koeficientų skirtumas. Atimamas prieš tai buvęs. --%
            dctcoef{i, j} (1, 1) = dctcoef{i, j} (1, 1) - prevdc;
            prevdc = temp;

            %--DC ir AC koeficientai matricioje nuskaitomi zigzago principu.--%
            dctcoef{i, j} = zigzag(dctcoef{i, j});

            nekoduotas = dctcoef{i, j};
            koduotas_RLE_MOD = RLE_mod(nekoduotas);

            l1 = l1 + length(koduotas_RLE_MOD);

            koduotas_AA = AA(dctcoef{i, j});
            l2 = l2 + length(koduotas_AA);

            koduotas_RLE = RLE(dctcoef{i, j});
            l3 = l3 + length(koduotas_RLE);
        end
    end

    size(1) = l1;
    size(2) = l2;
    size(3) = l3;

    %-- Paveiksliuko perrašymas į pradines struktūras. --%
    function img = jpginvtrans(dctcoef, xsize, ysize, xblocks, yblocks, qtable)
        prevdc = 0;

        %-- Einama per visus stulpelius ir eilutes (8x8 blokus). --%
        for j = 1 : xblocks
            for i = 1 : yblocks

                %-- DC ir AC koeficientai matricioje nuskaitomi zigzago principu iš priešingos pusės. --%
                dctcoef{i, j} = dezigzag(dctcoef{i, j});
            end
        end
    end
end

```

```

%-- Atstatomas DC koeficientų skirtumas. --%
dctcoef{i, j}(1, 1) = dctcoef{i, j}(1, 1) + prevdc;
prevdc=dctcoef{i, j}(1, 1);

%-- Gražinamos reikšmės prieš kvantavimą. --%
dctcoef{i, j} = dctcoef{i, j} .* qtable;

%--Atliekama atvirkštinė diskrečioji kosinuso transformacija (DCT). --%
imblock{i, j} = invdct(dctcoef{i, j});

%--Reikšmės iš [-128; 127] intervalo perslenkamos á [0; 255]. --%
imblock{i, j} = imblock{i, j} + 128;

%-- Sugražinamas 8x8 blokas. --%
img((i - 1) * 8 + 1 : i * 8, (j - 1) * 8 + 1 : j * 8) = imblock{i, j};
end
end

```

6.2. priedas. Pirmos dalies bandymų rezultatai

----- Eksperimentas Nr.1 Neglaudinti paveikslėlio duomenys: 147456 B RLE suglaudinti duomenys: 30498 B RLE modifikacija suglaudinti duomenys: 24662 B A&A suglaudinti duomenys: 24706 B -----	Neglaudinti paveikslėlio duomenys: 90000 B RLE suglaudinti duomenys: 21038 B RLE modifikacija suglaudinti duomenys: 17456 B A&A suglaudinti duomenys: 16210 B -----	RLE modifikacija suglaudinti duomenys: 376172 B A&A suglaudinti duomenys: 319813 B -----
Eksperimentas Nr.2 Neglaudinti paveikslėlio duomenys: 79800 B RLE suglaudinti duomenys: 18066 B RLE modifikacija suglaudinti duomenys: 16698 B A&A suglaudinti duomenys: 14240 B -----	Eksperimentas Nr.18 Neglaudinti paveikslėlio duomenys: 5760000 B RLE suglaudinti duomenys: 1325644 B RLE modifikacija suglaudinti duomenys: 1134216 B A&A suglaudinti duomenys: 989624 B -----	Eksperimentas Nr.34 Neglaudinti paveikslėlio duomenys: 4536000 B RLE suglaudinti duomenys: 810856 B RLE modifikacija suglaudinti duomenys: 726392 B A&A suglaudinti duomenys: 618924 B -----
Eksperimentas Nr.3 Neglaudinti paveikslėlio duomenys: 79800 B RLE suglaudinti duomenys: 14814 B RLE modifikacija suglaudinti duomenys: 12454 B A&A suglaudinti duomenys: 12590 B -----	Eksperimentas Nr.19 Neglaudinti paveikslėlio duomenys: 5760000 B RLE suglaudinti duomenys: 1141748 B RLE modifikacija suglaudinti duomenys: 984178 B A&A suglaudinti duomenys: 877269 B -----	Eksperimentas Nr.35 Neglaudinti paveikslėlio duomenys: 1083456 B RLE suglaudinti duomenys: 179226 B RLE modifikacija suglaudinti duomenys: 160424 B A&A suglaudinti duomenys: 140336 B -----
Eksperimentas Nr.4 Neglaudinti paveikslėlio duomenys: 79800 B RLE suglaudinti duomenys: 19778 B RLE modifikacija suglaudinti duomenys: 17606 B A&A suglaudinti duomenys: 15625 B -----	Eksperimentas Nr.20 Neglaudinti paveikslėlio duomenys: 90000 B RLE suglaudinti duomenys: 35738 B RLE modifikacija suglaudinti duomenys: 30682 B A&A suglaudinti duomenys: 25239 B -----	Eksperimentas Nr.36 Neglaudinti paveikslėlio duomenys: 1193157 B RLE suglaudinti duomenys: 143086 B RLE modifikacija suglaudinti duomenys: 127782 B A&A suglaudinti duomenys: 121015 B -----
Eksperimentas Nr.5 Neglaudinti paveikslėlio duomenys: 79800 B RLE suglaudinti duomenys: 30610 B RLE modifikacija suglaudinti duomenys: 27270 B A&A suglaudinti duomenys: 22299 B -----	Eksperimentas Nr.21 Neglaudinti paveikslėlio duomenys: 3932160 B RLE suglaudinti duomenys: 429620 B RLE modifikacija suglaudinti duomenys: 385274 B A&A suglaudinti duomenys: 380883 B -----	Eksperimentas Nr.37 Neglaudinti paveikslėlio duomenys: 456750 B RLE suglaudinti duomenys: 101156 B RLE modifikacija suglaudinti duomenys: 87030 B A&A suglaudinti duomenys: 75887 B -----
Eksperimentas Nr.6 Neglaudinti paveikslėlio duomenys: 79800 B RLE suglaudinti duomenys: 9390 B RLE modifikacija suglaudinti duomenys: 8204 B A&A suglaudinti duomenys: 8502 B -----	Eksperimentas Nr.22 Neglaudinti paveikslėlio duomenys: 3686400 B RLE suglaudinti duomenys: 385678 B RLE modifikacija suglaudinti duomenys: 336714 B A&A suglaudinti duomenys: 347888 B -----	Eksperimentas Nr.38 Neglaudinti paveikslėlio duomenys: 1122000 B RLE suglaudinti duomenys: 128608 B RLE modifikacija suglaudinti duomenys: 114002 B A&A suglaudinti duomenys: 110860 B -----
Eksperimentas Nr.7 Neglaudinti paveikslėlio duomenys: 79800 B RLE suglaudinti duomenys: 25342 B RLE modifikacija suglaudinti duomenys: 22618 B A&A suglaudinti duomenys: 19169 B -----	Eksperimentas Nr.23 Neglaudinti paveikslėlio duomenys: 708048 B RLE suglaudinti duomenys: 166158 B RLE modifikacija suglaudinti duomenys: 146644 B A&A suglaudinti duomenys: 125213 B -----	Eksperimentas Nr.39 Neglaudinti paveikslėlio duomenys: 165240 B RLE suglaudinti duomenys: 37514 B RLE modifikacija suglaudinti duomenys: 32232 B A&A suglaudinti duomenys: 27817 B -----
Eksperimentas Nr.8 Neglaudinti paveikslėlio duomenys: 79800 B RLE suglaudinti duomenys: 13848 B RLE modifikacija suglaudinti duomenys: 12388 B A&A suglaudinti duomenys: 11558 B -----	Eksperimentas Nr.24 Neglaudinti paveikslėlio duomenys: 150975 B RLE suglaudinti duomenys: 33666 B RLE modifikacija suglaudinti duomenys: 29668 B A&A suglaudinti duomenys: 25397 B -----	Eksperimentas Nr.40 Neglaudinti paveikslėlio duomenys: 1570020 B RLE suglaudinti duomenys: 285532 B RLE modifikacija suglaudinti duomenys: 256846 B A&A suglaudinti duomenys: 214579 B -----
Eksperimentas Nr.9 Neglaudinti paveikslėlio duomenys: 79800 B RLE suglaudinti duomenys: 16226 B RLE modifikacija suglaudinti duomenys: 14116 B A&A suglaudinti duomenys: 13697 B -----	Eksperimentas Nr.25 Neglaudinti paveikslėlio duomenys: 4320000 B RLE suglaudinti duomenys: 357730 B RLE modifikacija suglaudinti duomenys: 323642 B A&A suglaudinti duomenys: 348949 B -----	Eksperimentas Nr.41 Neglaudinti paveikslėlio duomenys: 4800000 B RLE suglaudinti duomenys: 798432 B RLE modifikacija suglaudinti duomenys: 711438 B A&A suglaudinti duomenys: 630657 B -----
Eksperimentas Nr.10 Neglaudinti paveikslėlio duomenys: 90000 B RLE suglaudinti duomenys: 14588 B RLE modifikacija suglaudinti duomenys: 12460 B A&A suglaudinti duomenys: 12232 B -----	Eksperimentas Nr.26 Neglaudinti paveikslėlio duomenys: 405000 B RLE suglaudinti duomenys: 19888 B RLE modifikacija suglaudinti duomenys: 17036 B A&A suglaudinti duomenys: 24338 B -----	Eksperimentas Nr.42 Neglaudinti paveikslėlio duomenys: 2359296 B RLE suglaudinti duomenys: 326818 B RLE modifikacija suglaudinti duomenys: 293624 B A&A suglaudinti duomenys: 266531 B -----
Eksperimentas Nr.11 Neglaudinti paveikslėlio duomenys: 90000 B RLE suglaudinti duomenys: 11178 B RLE modifikacija suglaudinti duomenys: 9546 B A&A suglaudinti duomenys: 9779 B -----	Eksperimentas Nr.27 Neglaudinti paveikslėlio duomenys: 150876 B RLE suglaudinti duomenys: 8928 B RLE modifikacija suglaudinti duomenys: 7440 B A&A suglaudinti duomenys: 8928 B -----	Eksperimentas Nr.43 Neglaudinti paveikslėlio duomenys: 3888000 B RLE suglaudinti duomenys: 708590 B RLE modifikacija suglaudinti duomenys: 620900 B A&A suglaudinti duomenys: 555968 B -----
Eksperimentas Nr.12 Neglaudinti paveikslėlio duomenys: 90000 B RLE suglaudinti duomenys: 24552 B RLE modifikacija suglaudinti duomenys: 21378 B A&A suglaudinti duomenys: 17959 B -----	Eksperimentas Nr.28 Neglaudinti paveikslėlio duomenys: 151320 B RLE suglaudinti duomenys: 20698 B RLE modifikacija suglaudinti duomenys: 18236 B A&A suglaudinti duomenys: 17614 B -----	Eksperimentas Nr.44 Neglaudinti paveikslėlio duomenys: 1200000 B RLE suglaudinti duomenys: 113234 B RLE modifikacija suglaudinti duomenys: 105244 B A&A suglaudinti duomenys: 101394 B -----
Eksperimentas Nr.13 Neglaudinti paveikslėlio duomenys: 90000 B RLE suglaudinti duomenys: 48438 B RLE modifikacija suglaudinti duomenys: 41824 B A&A suglaudinti duomenys: 34731 B -----	Eksperimentas Nr.29 Neglaudinti paveikslėlio duomenys: 151200 B RLE suglaudinti duomenys: 36274 B RLE modifikacija suglaudinti duomenys: 31182 B A&A suglaudinti duomenys: 27073 B -----	Eksperimentas Nr.45 Neglaudinti paveikslėlio duomenys: 3932160 B RLE suglaudinti duomenys: 801130 B RLE modifikacija suglaudinti duomenys: 702852 B A&A suglaudinti duomenys: 608854 B -----
Eksperimentas Nr.14 Neglaudinti paveikslėlio duomenys: 90000 B RLE suglaudinti duomenys: 25552 B RLE modifikacija suglaudinti duomenys: 20632 B A&A suglaudinti duomenys: 19459 B -----	Eksperimentas Nr.30 Neglaudinti paveikslėlio duomenys: 692640 B RLE suglaudinti duomenys: 88806 B RLE modifikacija suglaudinti duomenys: 73570 B A&A suglaudinti duomenys: 78407 B -----	Eksperimentas Nr.46 Neglaudinti paveikslėlio duomenys: 5760000 B RLE suglaudinti duomenys: 968464 B RLE modifikacija suglaudinti duomenys: 837636 B A&A suglaudinti duomenys: 778331 B -----
Eksperimentas Nr.15 Neglaudinti paveikslėlio duomenys: 90000 B RLE suglaudinti duomenys: 28354 B RLE modifikacija suglaudinti duomenys: 24158 B A&A suglaudinti duomenys: 20183 B -----	Eksperimentas Nr.31 Neglaudinti paveikslėlio duomenys: 2508462 B RLE suglaudinti duomenys: 462388 B RLE modifikacija suglaudinti duomenys: 417756 B A&A suglaudinti duomenys: 349480 B -----	Eksperimentas Nr.47 Neglaudinti paveikslėlio duomenys: 3147264 B RLE suglaudinti duomenys: 484554 B RLE modifikacija suglaudinti duomenys: 418414 B A&A suglaudinti duomenys: 402720 B -----
Eksperimentas Nr.16 Neglaudinti paveikslėlio duomenys: 90000 B RLE suglaudinti duomenys: 25394 B RLE modifikacija suglaudinti duomenys: 22808 B A&A suglaudinti duomenys: 18071 B -----	Eksperimentas Nr.32 Neglaudinti paveikslėlio duomenys: 2477466 B RLE suglaudinti duomenys: 399110 B RLE modifikacija suglaudinti duomenys: 361008 B A&A suglaudinti duomenys: 310751 B -----	Eksperimentas Nr.48 Neglaudinti paveikslėlio duomenys: 4800000 B RLE suglaudinti duomenys: 1241098 B RLE modifikacija suglaudinti duomenys: 1083256 B A&A suglaudinti duomenys: 905288 B -----
Eksperimentas Nr.17	Eksperimentas Nr.33 Neglaudinti paveikslėlio duomenys: 2477466 B RLE suglaudinti duomenys: 414148 B	Eksperimentas Nr.49 Neglaudinti paveikslėlio duomenys: 2088960 B RLE suglaudinti duomenys: 330746 B RLE modifikacija suglaudinti duomenys: 307510 B A&A suglaudinti duomenys: 265588 B

Eksperiments Nr.50 Neglaudinti paveikslēlio duomenys: 2359296 B RLE suglaudinti duomenys: 368882 B RLE modifikacija suglaudinti duomenys: 321870 B A&A suglaudinti duomenys: 300803 B	Eksperiments Nr.67 Neglaudinti paveikslēlio duomenys: 450858 B RLE suglaudinti duomenys: 55572 B RLE modifikacija suglaudinti duomenys: 47392 B A&A suglaudinti duomenys: 47724 B	Eksperiments Nr.84 Neglaudinti paveikslēlio duomenys: 699033 B RLE suglaudinti duomenys: 151198 B RLE modifikacija suglaudinti duomenys: 129518 B A&A suglaudinti duomenys: 112455 B
Eksperiments Nr.51 Neglaudinti paveikslēlio duomenys: 1157238 B RLE suglaudinti duomenys: 176452 B RLE modifikacija suglaudinti duomenys: 148384 B A&A suglaudinti duomenys: 143577 B	Eksperiments Nr.68 Neglaudinti paveikslēlio duomenys: 469404 B RLE suglaudinti duomenys: 115348 B RLE modifikacija suglaudinti duomenys: 94682 B A&A suglaudinti duomenys: 84627 B	Eksperiments Nr.85 Neglaudinti paveikslēlio duomenys: 562500 B RLE suglaudinti duomenys: 162892 B RLE modifikacija suglaudinti duomenys: 141048 B A&A suglaudinti duomenys: 115974 B
Eksperiments Nr.52 Neglaudinti paveikslēlio duomenys: 2073600 B RLE suglaudinti duomenys: 571004 B RLE modifikacija suglaudinti duomenys: 497564 B A&A suglaudinti duomenys: 417352 B	Eksperiments Nr.69 Neglaudinti paveikslēlio duomenys: 721800 B RLE suglaudinti duomenys: 105186 B RLE modifikacija suglaudinti duomenys: 87842 B A&A suglaudinti duomenys: 89026 B	Eksperiments Nr.86 Neglaudinti paveikslēlio duomenys: 1440000 B RLE suglaudinti duomenys: 327108 B RLE modifikacija suglaudinti duomenys: 282514 B A&A suglaudinti duomenys: 244116 B
Eksperiments Nr.53 Neglaudinti paveikslēlio duomenys: 718200 B RLE suglaudinti duomenys: 90926 B RLE modifikacija suglaudinti duomenys: 82900 B A&A suglaudinti duomenys: 74675 B	Eksperiments Nr.70 Neglaudinti paveikslēlio duomenys: 524724 B RLE suglaudinti duomenys: 76532 B RLE modifikacija suglaudinti duomenys: 66866 B A&A suglaudinti duomenys: 62445 B	Eksperiments Nr.87 Neglaudinti paveikslēlio duomenys: 968100 B RLE suglaudinti duomenys: 217042 B RLE modifikacija suglaudinti duomenys: 190354 B A&A suglaudinti duomenys: 165679 B
Eksperiments Nr.54 Neglaudinti paveikslēlio duomenys: 1005600 B RLE suglaudinti duomenys: 152652 B RLE modifikacija suglaudinti duomenys: 132116 B A&A suglaudinti duomenys: 122498 B	Eksperiments Nr.71 Neglaudinti paveikslēlio duomenys: 499500 B RLE suglaudinti duomenys: 66410 B RLE modifikacija suglaudinti duomenys: 57358 B A&A suglaudinti duomenys: 57285 B	Eksperiments Nr.88 Neglaudinti paveikslēlio duomenys: 608652 B RLE suglaudinti duomenys: 177086 B RLE modifikacija suglaudinti duomenys: 152598 B A&A suglaudinti duomenys: 125874 B
Eksperiments Nr.55 Neglaudinti paveikslēlio duomenys: 1966080 B RLE suglaudinti duomenys: 288564 B RLE modifikacija suglaudinti duomenys: 252756 B A&A suglaudinti duomenys: 229741 B	Eksperiments Nr.72 Neglaudinti paveikslēlio duomenys: 716943 B RLE suglaudinti duomenys: 48414 B RLE modifikacija suglaudinti duomenys: 44984 B A&A suglaudinti duomenys: 49799 B	Eksperiments Nr.89 Neglaudinti paveikslēlio duomenys: 1166400 B RLE suglaudinti duomenys: 265452 B RLE modifikacija suglaudinti duomenys: 220208 B A&A suglaudinti duomenys: 207723 B
Eksperiments Nr.56 Neglaudinti paveikslēlio duomenys: 790020 B RLE suglaudinti duomenys: 193870 B RLE modifikacija suglaudinti duomenys: 168292 B A&A suglaudinti duomenys: 143869 B	Eksperiments Nr.73 Neglaudinti paveikslēlio duomenys: 275520 B RLE suglaudinti duomenys: 40378 B RLE modifikacija suglaudinti duomenys: 35156 B A&A suglaudinti duomenys: 33441 B	Eksperiments Nr.90 Neglaudinti paveikslēlio duomenys: 583200 B RLE suglaudinti duomenys: 118374 B RLE modifikacija suglaudinti duomenys: 101036 B A&A suglaudinti duomenys: 87576 B
Eksperiments Nr.57 Neglaudinti paveikslēlio duomenys: 2359296 B RLE suglaudinti duomenys: 441952 B RLE modifikacija suglaudinti duomenys: 389652 B A&A suglaudinti duomenys: 339022 B	Eksperiments Nr.74 Neglaudinti paveikslēlio duomenys: 443625 B RLE suglaudinti duomenys: 40652 B RLE modifikacija suglaudinti duomenys: 37320 B A&A suglaudinti duomenys: 37604 B	Eksperiments Nr.91 Neglaudinti paveikslēlio duomenys: 633360 B RLE suglaudinti duomenys: 88158 B RLE modifikacija suglaudinti duomenys: 75846 B A&A suglaudinti duomenys: 72001 B
Eksperiments Nr.58 Neglaudinti paveikslēlio duomenys: 562500 B RLE suglaudinti duomenys: 131734 B RLE modifikacija suglaudinti duomenys: 112388 B A&A suglaudinti duomenys: 100117 B	Eksperiments Nr.75 Neglaudinti paveikslēlio duomenys: 1531200 B RLE suglaudinti duomenys: 179716 B RLE modifikacija suglaudinti duomenys: 159616 B A&A suglaudinti duomenys: 157498 B	Eksperiments Nr.92 Neglaudinti paveikslēlio duomenys: 402300 B RLE suglaudinti duomenys: 124930 B RLE modifikacija suglaudinti duomenys: 106652 B A&A suglaudinti duomenys: 90619 B
Eksperiments Nr.59 Neglaudinti paveikslēlio duomenys: 6912000 B RLE suglaudinti duomenys: 467420 B RLE modifikacija suglaudinti duomenys: 424580 B A&A suglaudinti duomenys: 463899 B	Eksperiments Nr.76 Neglaudinti paveikslēlio duomenys: 374850 B RLE suglaudinti duomenys: 39950 B RLE modifikacija suglaudinti duomenys: 35514 B A&A suglaudinti duomenys: 35534 B	Eksperiments Nr.93 Neglaudinti paveikslēlio duomenys: 568980 B RLE suglaudinti duomenys: 101004 B RLE modifikacija suglaudinti duomenys: 87952 B A&A suglaudinti duomenys: 76824 B
Eksperiments Nr.60 Neglaudinti paveikslēlio duomenys: 1822500 B RLE suglaudinti duomenys: 284350 B RLE modifikacija suglaudinti duomenys: 250142 B A&A suglaudinti duomenys: 227011 B	Eksperiments Nr.77 Neglaudinti paveikslēlio duomenys: 243000 B RLE suglaudinti duomenys: 57386 B RLE modifikacija suglaudinti duomenys: 48766 B A&A suglaudinti duomenys: 43251 B	Eksperiments Nr.94 Neglaudinti paveikslēlio duomenys: 593616 B RLE suglaudinti duomenys: 123916 B RLE modifikacija suglaudinti duomenys: 108860 B A&A suglaudinti duomenys: 91788 B
Eksperiments Nr.61 Neglaudinti paveikslēlio duomenys: 867825 B RLE suglaudinti duomenys: 140556 B RLE modifikacija suglaudinti duomenys: 120400 B A&A suglaudinti duomenys: 113092 B	Eksperiments Nr.78 Neglaudinti paveikslēlio duomenys: 227853 B RLE suglaudinti duomenys: 43392 B RLE modifikacija suglaudinti duomenys: 38384 B A&A suglaudinti duomenys: 32940 B	Eksperiments Nr.95 Neglaudinti paveikslēlio duomenys: 562500 B RLE suglaudinti duomenys: 93576 B RLE modifikacija suglaudinti duomenys: 82466 B A&A suglaudinti duomenys: 73701 B
Eksperiments Nr.62 Neglaudinti paveikslēlio duomenys: 4962045 B RLE suglaudinti duomenys: 351378 B RLE modifikacija suglaudinti duomenys: 310394 B A&A suglaudinti duomenys: 361621 B	Eksperiments Nr.79 Neglaudinti paveikslēlio duomenys: 442260 B RLE suglaudinti duomenys: 82740 B RLE modifikacija suglaudinti duomenys: 67036 B A&A suglaudinti duomenys: 66832 B	Eksperiments Nr.96 Neglaudinti paveikslēlio duomenys: 630000 B RLE suglaudinti duomenys: 125718 B RLE modifikacija suglaudinti duomenys: 108112 B A&A suglaudinti duomenys: 98481 B
Eksperiments Nr.63 Neglaudinti paveikslēlio duomenys: 944640 B RLE suglaudinti duomenys: 73098 B RLE modifikacija suglaudinti duomenys: 67268 B A&A suglaudinti duomenys: 72883 B	Eksperiments Nr.80 Neglaudinti paveikslēlio duomenys: 282240 B RLE suglaudinti duomenys: 51960 B RLE modifikacija suglaudinti duomenys: 45300 B A&A suglaudinti duomenys: 40793 B	Eksperiments Nr.97 Neglaudinti paveikslēlio duomenys: 218610 B RLE suglaudinti duomenys: 43162 B RLE modifikacija suglaudinti duomenys: 37508 B A&A suglaudinti duomenys: 32707 B
Eksperiments Nr.64 Neglaudinti paveikslēlio duomenys: 988812 B RLE suglaudinti duomenys: 72142 B RLE modifikacija suglaudinti duomenys: 67328 B A&A suglaudinti duomenys: 72210 B	Eksperiments Nr.81 Neglaudinti paveikslēlio duomenys: 493050 B RLE suglaudinti duomenys: 74776 B RLE modifikacija suglaudinti duomenys: 64112 B A&A suglaudinti duomenys: 59129 B	Eksperiments Nr.98 Neglaudinti paveikslēlio duomenys: 745500 B RLE suglaudinti duomenys: 128424 B RLE modifikacija suglaudinti duomenys: 115800 B A&A suglaudinti duomenys: 102684 B
Eksperiments Nr.65 Neglaudinti paveikslēlio duomenys: 780669 B RLE suglaudinti duomenys: 122880 B RLE modifikacija suglaudinti duomenys: 101564 B A&A suglaudinti duomenys: 99485 B	Eksperiments Nr.82 Neglaudinti paveikslēlio duomenys: 134250 B RLE suglaudinti duomenys: 36994 B RLE modifikacija suglaudinti duomenys: 31290 B A&A suglaudinti duomenys: 26647 B	Eksperiments Nr.99 Neglaudinti paveikslēlio duomenys: 204480 B RLE suglaudinti duomenys: 52060 B RLE modifikacija suglaudinti duomenys: 439918 B A&A suglaudinti duomenys: 37630 B
Eksperiments Nr.66 Neglaudinti paveikslēlio duomenys: 430080 B RLE suglaudinti duomenys: 74774 B RLE modifikacija suglaudinti duomenys: 63884 B A&A suglaudinti duomenys: 60501 B	Eksperiments Nr.83 Neglaudinti paveikslēlio duomenys: 773760 B RLE suglaudinti duomenys: 227616 B RLE modifikacija suglaudinti duomenys: 206152 B A&A suglaudinti duomenys: 170280 B	Eksperiments Nr.100 Neglaudinti paveikslēlio duomenys: 2098176 B RLE suglaudinti duomenys: 502154 B RLE modifikacija suglaudinti duomenys: 439918 B A&A suglaudinti duomenys: 371894 B

6.3. priedas. Antros dalies bandymų rezultatai

<p>-----</p> <p>Eksperimentas Nr.1</p> <p>Neglaudinti paveikslėlio duomenys: 147456 B</p> <p>RLE suglaudinti duomenys: 46404 B</p> <p>RLE modifikacija suglaudinti duomenys: 34846 B</p> <p>A&A suglaudinti duomenys: 36548 B</p> <p>-----</p> <p>Eksperimentas Nr.2</p> <p>Neglaudinti paveikslėlio duomenys: 79800 B</p> <p>RLE suglaudinti duomenys: 19210 B</p> <p>RLE modifikacija suglaudinti duomenys: 16872 B</p> <p>A&A suglaudinti duomenys: 14369 B</p> <p>-----</p> <p>Eksperimentas Nr.3</p> <p>Neglaudinti paveikslėlio duomenys: 79800 B</p> <p>RLE suglaudinti duomenys: 15524 B</p> <p>RLE modifikacija suglaudinti duomenys: 12712 B</p> <p>A&A suglaudinti duomenys: 12699 B</p> <p>-----</p> <p>Eksperimentas Nr.4</p> <p>Neglaudinti paveikslėlio duomenys: 79800 B</p> <p>RLE suglaudinti duomenys: 21062 B</p> <p>RLE modifikacija suglaudinti duomenys: 17892 B</p> <p>A&A suglaudinti duomenys: 15738 B</p> <p>-----</p> <p>Eksperimentas Nr.5</p> <p>Neglaudinti paveikslėlio duomenys: 79800 B</p> <p>RLE suglaudinti duomenys: 32120 B</p> <p>RLE modifikacija suglaudinti duomenys: 27558 B</p> <p>A&A suglaudinti duomenys: 22423 B</p> <p>-----</p> <p>Eksperimentas Nr.6</p> <p>Neglaudinti paveikslėlio duomenys: 79800 B</p> <p>RLE suglaudinti duomenys: 9878 B</p> <p>RLE modifikacija suglaudinti duomenys: 8432 B</p> <p>A&A suglaudinti duomenys: 8587 B</p> <p>-----</p> <p>Eksperimentas Nr.7</p> <p>Neglaudinti paveikslėlio duomenys: 79800 B</p> <p>RLE suglaudinti duomenys: 26910 B</p> <p>RLE modifikacija suglaudinti duomenys: 22966 B</p> <p>A&A suglaudinti duomenys: 19318 B</p> <p>-----</p> <p>Eksperimentas Nr.8</p> <p>Neglaudinti paveikslėlio duomenys: 79800 B</p> <p>RLE suglaudinti duomenys: 14856 B</p> <p>RLE modifikacija suglaudinti duomenys: 12730 B</p> <p>A&A suglaudinti duomenys: 11758 B</p> <p>-----</p> <p>Eksperimentas Nr.9</p> <p>Neglaudinti paveikslėlio duomenys: 79800 B</p> <p>RLE suglaudinti duomenys: 17158 B</p> <p>RLE modifikacija suglaudinti duomenys: 14322 B</p> <p>A&A suglaudinti duomenys: 13844 B</p> <p>-----</p> <p>Eksperimentas Nr.10</p> <p>Neglaudinti paveikslėlio duomenys: 90000 B</p> <p>RLE suglaudinti duomenys: 15164 B</p> <p>RLE modifikacija suglaudinti duomenys: 12680 B</p> <p>A&A suglaudinti duomenys: 12320 B</p> <p>-----</p> <p>Eksperimentas Nr.11</p> <p>Neglaudinti paveikslėlio duomenys: 90000 B</p> <p>RLE suglaudinti duomenys: 11590 B</p> <p>RLE modifikacija suglaudinti duomenys: 9698 B</p> <p>A&A suglaudinti duomenys: 9837 B</p> <p>-----</p> <p>Eksperimentas Nr.12</p> <p>Neglaudinti paveikslėlio duomenys: 90000 B</p> <p>RLE suglaudinti duomenys: 25172 B</p> <p>RLE modifikacija suglaudinti duomenys: 21534 B</p> <p>A&A suglaudinti duomenys: 18028 B</p> <p>-----</p> <p>Eksperimentas Nr.13</p> <p>Neglaudinti paveikslėlio duomenys: 90000 B</p> <p>RLE suglaudinti duomenys: 51436 B</p> <p>RLE modifikacija suglaudinti duomenys: 42596 B</p> <p>A&A suglaudinti duomenys: 34859 B</p> <p>-----</p> <p>Eksperimentas Nr.14</p> <p>Neglaudinti paveikslėlio duomenys: 90000 B</p> <p>RLE suglaudinti duomenys: 26042 B</p> <p>RLE modifikacija suglaudinti duomenys: 20798 B</p> <p>A&A suglaudinti duomenys: 19522 B</p> <p>-----</p> <p>Eksperimentas Nr.15</p> <p>Neglaudinti paveikslėlio duomenys: 90000 B</p> <p>RLE suglaudinti duomenys: 28838 B</p> <p>RLE modifikacija suglaudinti duomenys: 24234 B</p> <p>A&A suglaudinti duomenys: 20212 B</p> <p>-----</p> <p>Eksperimentas Nr.16</p> <p>Neglaudinti paveikslėlio duomenys: 90000 B</p> <p>RLE suglaudinti duomenys: 26074 B</p> <p>RLE modifikacija suglaudinti duomenys: 23042 B</p> <p>A&A suglaudinti duomenys: 18179 B</p> <p>-----</p> <p>Eksperimentas Nr.17</p> <p>Neglaudinti paveikslėlio duomenys: 90000 B</p> <p>RLE suglaudinti duomenys: 21564 B</p>	<p>RLE modifikacija suglaudinti duomenys: 17570 B</p> <p>A&A suglaudinti duomenys: 16257 B</p> <p>-----</p> <p>Eksperimentas Nr.18</p> <p>Neglaudinti paveikslėlio duomenys: 5760000 B</p> <p>RLE suglaudinti duomenys: 1922962 B</p> <p>RLE modifikacija suglaudinti duomenys: 1716956 B</p> <p>A&A suglaudinti duomenys: 1341004 B</p> <p>-----</p> <p>Eksperimentas Nr.19</p> <p>Neglaudinti paveikslėlio duomenys: 5760000 B</p> <p>RLE suglaudinti duomenys: 1730432 B</p> <p>RLE modifikacija suglaudinti duomenys: 1581014 B</p> <p>A&A suglaudinti duomenys: 1237218 B</p> <p>-----</p> <p>Eksperimentas Nr.20</p> <p>Neglaudinti paveikslėlio duomenys: 90000 B</p> <p>RLE suglaudinti duomenys: 36748 B</p> <p>RLE modifikacija suglaudinti duomenys: 30832 B</p> <p>A&A suglaudinti duomenys: 25277 B</p> <p>-----</p> <p>Eksperimentas Nr.21</p> <p>Neglaudinti paveikslėlio duomenys: 3932160 B</p> <p>RLE suglaudinti duomenys: 587902 B</p> <p>RLE modifikacija suglaudinti duomenys: 529178 B</p> <p>A&A suglaudinti duomenys: 479856 B</p> <p>-----</p> <p>Eksperimentas Nr.22</p> <p>Neglaudinti paveikslėlio duomenys: 3686400 B</p> <p>RLE suglaudinti duomenys: 649320 B</p> <p>RLE modifikacija suglaudinti duomenys: 579556 B</p> <p>A&A suglaudinti duomenys: 529014 B</p> <p>-----</p> <p>Eksperimentas Nr.23</p> <p>Neglaudinti paveikslėlio duomenys: 708048 B</p> <p>RLE suglaudinti duomenys: 188674 B</p> <p>RLE modifikacija suglaudinti duomenys: 159710 B</p> <p>A&A suglaudinti duomenys: 136281 B</p> <p>-----</p> <p>Eksperimentas Nr.24</p> <p>Neglaudinti paveikslėlio duomenys: 150975 B</p> <p>RLE suglaudinti duomenys: 39894 B</p> <p>RLE modifikacija suglaudinti duomenys: 33372 B</p> <p>A&A suglaudinti duomenys: 29198 B</p> <p>-----</p> <p>Eksperimentas Nr.25</p> <p>Neglaudinti paveikslėlio duomenys: 4320000 B</p> <p>RLE suglaudinti duomenys: 463412 B</p> <p>RLE modifikacija suglaudinti duomenys: 413628 B</p> <p>A&A suglaudinti duomenys: 412217 B</p> <p>-----</p> <p>Eksperimentas Nr.26</p> <p>Neglaudinti paveikslėlio duomenys: 405000 B</p> <p>RLE suglaudinti duomenys: 21710 B</p> <p>RLE modifikacija suglaudinti duomenys: 18328 B</p> <p>A&A suglaudinti duomenys: 25865 B</p> <p>-----</p> <p>Eksperimentas Nr.27</p> <p>Neglaudinti paveikslėlio duomenys: 150876 B</p> <p>RLE suglaudinti duomenys: 8928 B</p> <p>RLE modifikacija suglaudinti duomenys: 7440 B</p> <p>A&A suglaudinti duomenys: 8928 B</p> <p>-----</p> <p>Eksperimentas Nr.28</p> <p>Neglaudinti paveikslėlio duomenys: 151320 B</p> <p>RLE suglaudinti duomenys: 21286 B</p> <p>RLE modifikacija suglaudinti duomenys: 18382 B</p> <p>A&A suglaudinti duomenys: 17676 B</p> <p>-----</p> <p>Eksperimentas Nr.29</p> <p>Neglaudinti paveikslėlio duomenys: 151200 B</p> <p>RLE suglaudinti duomenys: 50932 B</p> <p>RLE modifikacija suglaudinti duomenys: 42360 B</p> <p>A&A suglaudinti duomenys: 36470 B</p> <p>-----</p> <p>Eksperimentas Nr.30</p> <p>Neglaudinti paveikslėlio duomenys: 692640 B</p> <p>RLE suglaudinti duomenys: 116532 B</p> <p>RLE modifikacija suglaudinti duomenys: 96678 B</p> <p>A&A suglaudinti duomenys: 94074 B</p> <p>-----</p> <p>Eksperimentas Nr.31</p> <p>Neglaudinti paveikslėlio duomenys: 2508462 B</p> <p>RLE suglaudinti duomenys: 566402 B</p> <p>RLE modifikacija suglaudinti duomenys: 516804 B</p> <p>A&A suglaudinti duomenys: 402212 B</p> <p>-----</p> <p>Eksperimentas Nr.32</p> <p>Neglaudinti paveikslėlio duomenys: 2477466 B</p> <p>RLE suglaudinti duomenys: 487770 B</p> <p>RLE modifikacija suglaudinti duomenys: 445610 B</p> <p>A&A suglaudinti duomenys: 356292 B</p> <p>-----</p> <p>Eksperimentas Nr.33</p> <p>Neglaudinti paveikslėlio duomenys: 2477466 B</p> <p>RLE suglaudinti duomenys: 502588 B</p> <p>RLE modifikacija suglaudinti duomenys: 462738 B</p> <p>A&A suglaudinti duomenys: 364204 B</p> <p>-----</p> <p>Eksperimentas Nr.34</p>	<p>Neglaudinti paveikslėlio duomenys: 4536000 B</p> <p>RLE suglaudinti duomenys: 999130 B</p> <p>RLE modifikacija suglaudinti duomenys: 902350 B</p> <p>A&A suglaudinti duomenys: 720042 B</p> <p>-----</p> <p>Eksperimentas Nr.35</p> <p>Neglaudinti paveikslėlio duomenys: 1083456 B</p> <p>RLE suglaudinti duomenys: 216990 B</p> <p>RLE modifikacija suglaudinti duomenys: 198902 B</p> <p>A&A suglaudinti duomenys: 157205 B</p> <p>-----</p> <p>Eksperimentas Nr.36</p> <p>Neglaudinti paveikslėlio duomenys: 1193157 B</p> <p>RLE suglaudinti duomenys: 175586 B</p> <p>RLE modifikacija suglaudinti duomenys: 159136 B</p> <p>A&A suglaudinti duomenys: 137291 B</p> <p>-----</p> <p>Eksperimentas Nr.37</p> <p>Neglaudinti paveikslėlio duomenys: 456750 B</p> <p>RLE suglaudinti duomenys: 135104 B</p> <p>RLE modifikacija suglaudinti duomenys: 118648 B</p> <p>A&A suglaudinti duomenys: 93087 B</p> <p>-----</p> <p>Eksperimentas Nr.38</p> <p>Neglaudinti paveikslėlio duomenys: 1122000 B</p> <p>RLE suglaudinti duomenys: 164820 B</p> <p>RLE modifikacija suglaudinti duomenys: 146288 B</p> <p>A&A suglaudinti duomenys: 131580 B</p> <p>-----</p> <p>Eksperimentas Nr.39</p> <p>Neglaudinti paveikslėlio duomenys: 165240 B</p> <p>RLE suglaudinti duomenys: 47426 B</p> <p>RLE modifikacija suglaudinti duomenys: 41386 B</p> <p>A&A suglaudinti duomenys: 33364 B</p> <p>-----</p> <p>Eksperimentas Nr.40</p> <p>Neglaudinti paveikslėlio duomenys: 1570020 B</p> <p>RLE suglaudinti duomenys: 335470 B</p> <p>RLE modifikacija suglaudinti duomenys: 308794 B</p> <p>A&A suglaudinti duomenys: 235520 B</p> <p>-----</p> <p>Eksperimentas Nr.41</p> <p>Neglaudinti paveikslėlio duomenys: 4800000 B</p> <p>RLE suglaudinti duomenys: 1102412 B</p> <p>RLE modifikacija suglaudinti duomenys: 950798 B</p> <p>A&A suglaudinti duomenys: 817570 B</p> <p>-----</p> <p>Eksperimentas Nr.42</p> <p>Neglaudinti paveikslėlio duomenys: 2359296 B</p> <p>RLE suglaudinti duomenys: 385816 B</p> <p>RLE modifikacija suglaudinti duomenys: 340054 B</p> <p>A&A suglaudinti duomenys: 297166 B</p> <p>-----</p> <p>Eksperimentas Nr.43</p> <p>Neglaudinti paveikslėlio duomenys: 3888000 B</p> <p>RLE suglaudinti duomenys: 996716 B</p> <p>RLE modifikacija suglaudinti duomenys: 859356 B</p> <p>A&A suglaudinti duomenys: 728065 B</p> <p>-----</p> <p>Eksperimentas Nr.44</p> <p>Neglaudinti paveikslėlio duomenys: 1200000 B</p> <p>RLE suglaudinti duomenys: 153358 B</p> <p>RLE modifikacija suglaudinti duomenys: 140608 B</p> <p>A&A suglaudinti duomenys: 125422 B</p> <p>-----</p> <p>Eksperimentas Nr.45</p> <p>Neglaudinti paveikslėlio duomenys: 3932160 B</p> <p>RLE suglaudinti duomenys: 1055218 B</p> <p>RLE modifikacija suglaudinti duomenys: 936066 B</p> <p>A&A suglaudinti duomenys: 751151 B</p> <p>-----</p> <p>Eksperimentas Nr.46</p> <p>Neglaudinti paveikslėlio duomenys: 5760000 B</p> <p>RLE suglaudinti duomenys: 1525186 B</p> <p>RLE modifikacija suglaudinti duomenys: 1363004 B</p> <p>A&A suglaudinti duomenys: 1124700 B</p> <p>-----</p> <p>Eksperimentas Nr.47</p> <p>Neglaudinti paveikslėlio duomenys: 3147264 B</p> <p>RLE suglaudinti duomenys: 594076 B</p> <p>RLE modifikacija suglaudinti duomenys: 520704 B</p> <p>A&A suglaudinti duomenys: 458451 B</p> <p>-----</p> <p>Eksperimentas Nr.48</p> <p>Neglaudinti paveikslėlio duomenys: 4800000 B</p> <p>RLE suglaudinti duomenys: 1550246 B</p> <p>RLE modifikacija suglaudinti duomenys: 1342554 B</p> <p>A&A suglaudinti duomenys: 1088921 B</p> <p>-----</p> <p>Eksperimentas Nr.49</p> <p>Neglaudinti paveikslėlio duomenys: 2088960 B</p> <p>RLE suglaudinti duomenys: 371046 B</p> <p>RLE modifikacija suglaudinti duomenys: 333620 B</p> <p>A&A suglaudinti duomenys: 282090 B</p> <p>-----</p> <p>Eksperimentas Nr.50</p> <p>Neglaudinti paveikslėlio duomenys: 2359296 B</p> <p>RLE suglaudinti duomenys: 446096 B</p> <p>RLE modifikacija suglaudinti duomenys: 400716 B</p> <p>A&A suglaudinti duomenys: 340115 B</p>
---	--	--

Eksperimentas Nr.51 Neglaudinti paveikslėlio duomenys: 1157238 B RLE suglaudinti duomenys: 228118 B RLE modifikacija suglaudinti duomenys: 195626 B A&A suglaudinti duomenys: 172276 B	A&A suglaudinti duomenys: 50882 B	RLE modifikacija suglaudinti duomenys: 178946 B A&A suglaudinti duomenys: 144559 B
Eksperimentas Nr.52 Neglaudinti paveikslėlio duomenys: 2073600 B RLE suglaudinti duomenys: 909430 B RLE modifikacija suglaudinti duomenys: 816034 B A&A suglaudinti duomenys: 631573 B	Eksperimentas Nr.68 Neglaudinti paveikslėlio duomenys: 469404 B RLE suglaudinti duomenys: 138182 B RLE modifikacija suglaudinti duomenys: 115394 B A&A suglaudinti duomenys: 91739 B	Eksperimentas Nr.85 Neglaudinti paveikslėlio duomenys: 562500 B RLE suglaudinti duomenys: 249924 B RLE modifikacija suglaudinti duomenys: 225974 B A&A suglaudinti duomenys: 167170 B
Eksperimentas Nr.53 Neglaudinti paveikslėlio duomenys: 718200 B RLE suglaudinti duomenys: 122658 B RLE modifikacija suglaudinti duomenys: 110556 B A&A suglaudinti duomenys: 93794 B	Eksperimentas Nr.69 Neglaudinti paveikslėlio duomenys: 721800 B RLE suglaudinti duomenys: 123792 B RLE modifikacija suglaudinti duomenys: 105336 B A&A suglaudinti duomenys: 99792 B	Eksperimentas Nr.86 Neglaudinti paveikslėlio duomenys: 1440000 B RLE suglaudinti duomenys: 385836 B RLE modifikacija suglaudinti duomenys: 321132 B A&A suglaudinti duomenys: 278806 B
Eksperimentas Nr.54 Neglaudinti paveikslėlio duomenys: 1005600 B RLE suglaudinti duomenys: 253928 B RLE modifikacija suglaudinti duomenys: 226520 B A&A suglaudinti duomenys: 189678 B	Eksperimentas Nr.70 Neglaudinti paveikslėlio duomenys: 524724 B RLE suglaudinti duomenys: 88584 B RLE modifikacija suglaudinti duomenys: 77592 B A&A suglaudinti duomenys: 68474 B	Eksperimentas Nr.87 Neglaudinti paveikslėlio duomenys: 968100 B RLE suglaudinti duomenys: 236162 B RLE modifikacija suglaudinti duomenys: 199460 B A&A suglaudinti duomenys: 171535 B
Eksperimentas Nr.55 Neglaudinti paveikslėlio duomenys: 1966080 B RLE suglaudinti duomenys: 386990 B RLE modifikacija suglaudinti duomenys: 333838 B A&A suglaudinti duomenys: 286470 B	Eksperimentas Nr.71 Neglaudinti paveikslėlio duomenys: 499500 B RLE suglaudinti duomenys: 88812 B RLE modifikacija suglaudinti duomenys: 76114 B A&A suglaudinti duomenys: 72413 B	Eksperimentas Nr.88 Neglaudinti paveikslėlio duomenys: 608652 B RLE suglaudinti duomenys: 220752 B RLE modifikacija suglaudinti duomenys: 195696 B A&A suglaudinti duomenys: 149276 B
Eksperimentas Nr.56 Neglaudinti paveikslėlio duomenys: 790020 B RLE suglaudinti duomenys: 265110 B RLE modifikacija suglaudinti duomenys: 228978 B A&A suglaudinti duomenys: 185212 B	Eksperimentas Nr.72 Neglaudinti paveikslėlio duomenys: 716943 B RLE suglaudinti duomenys: 55006 B RLE modifikacija suglaudinti duomenys: 51292 B A&A suglaudinti duomenys: 53232 B	Eksperimentas Nr.89 Neglaudinti paveikslėlio duomenys: 1166400 B RLE suglaudinti duomenys: 383476 B RLE modifikacija suglaudinti duomenys: 335348 B A&A suglaudinti duomenys: 277570 B
Eksperimentas Nr.57 Neglaudinti paveikslėlio duomenys: 2359296 B RLE suglaudinti duomenys: 517462 B RLE modifikacija suglaudinti duomenys: 454478 B A&A suglaudinti duomenys: 373718 B	Eksperimentas Nr.73 Neglaudinti paveikslėlio duomenys: 275520 B RLE suglaudinti duomenys: 48460 B RLE modifikacija suglaudinti duomenys: 42614 B A&A suglaudinti duomenys: 37820 B	Eksperimentas Nr.90 Neglaudinti paveikslėlio duomenys: 583200 B RLE suglaudinti duomenys: 184362 B RLE modifikacija suglaudinti duomenys: 162626 B A&A suglaudinti duomenys: 128700 B
Eksperimentas Nr.58 Neglaudinti paveikslėlio duomenys: 562500 B RLE suglaudinti duomenys: 151004 B RLE modifikacija suglaudinti duomenys: 131780 B A&A suglaudinti duomenys: 107652 B	Eksperimentas Nr.74 Neglaudinti paveikslėlio duomenys: 443625 B RLE suglaudinti duomenys: 49746 B RLE modifikacija suglaudinti duomenys: 45512 B A&A suglaudinti duomenys: 42865 B	Eksperimentas Nr.91 Neglaudinti paveikslėlio duomenys: 633360 B RLE suglaudinti duomenys: 133450 B RLE modifikacija suglaudinti duomenys: 115538 B A&A suglaudinti duomenys: 99467 B
Eksperimentas Nr.59 Neglaudinti paveikslėlio duomenys: 6912000 B RLE suglaudinti duomenys: 619924 B RLE modifikacija suglaudinti duomenys: 538450 B A&A suglaudinti duomenys: 549609 B	Eksperimentas Nr.75 Neglaudinti paveikslėlio duomenys: 1531200 B RLE suglaudinti duomenys: 216698 B RLE modifikacija suglaudinti duomenys: 194118 B A&A suglaudinti duomenys: 178455 B	Eksperimentas Nr.92 Neglaudinti paveikslėlio duomenys: 402300 B RLE suglaudinti duomenys: 145710 B RLE modifikacija suglaudinti duomenys: 126948 B A&A suglaudinti duomenys: 100123 B
Eksperimentas Nr.60 Neglaudinti paveikslėlio duomenys: 1822500 B RLE suglaudinti duomenys: 385748 B RLE modifikacija suglaudinti duomenys: 335390 B A&A suglaudinti duomenys: 285224 B	Eksperimentas Nr.76 Neglaudinti paveikslėlio duomenys: 374850 B RLE suglaudinti duomenys: 49548 B RLE modifikacija suglaudinti duomenys: 44090 B A&A suglaudinti duomenys: 41293 B	Eksperimentas Nr.93 Neglaudinti paveikslėlio duomenys: 568980 B RLE suglaudinti duomenys: 151554 B RLE modifikacija suglaudinti duomenys: 135940 B A&A suglaudinti duomenys: 107069 B
Eksperimentas Nr.61 Neglaudinti paveikslėlio duomenys: 867825 B RLE suglaudinti duomenys: 171720 B RLE modifikacija suglaudinti duomenys: 150092 B A&A suglaudinti duomenys: 129295 B	Eksperimentas Nr.77 Neglaudinti paveikslėlio duomenys: 243000 B RLE suglaudinti duomenys: 67842 B RLE modifikacija suglaudinti duomenys: 58904 B A&A suglaudinti duomenys: 47874 B	Eksperimentas Nr.94 Neglaudinti paveikslėlio duomenys: 593616 B RLE suglaudinti duomenys: 181100 B RLE modifikacija suglaudinti duomenys: 165288 B A&A suglaudinti duomenys: 127520 B
Eksperimentas Nr.62 Neglaudinti paveikslėlio duomenys: 4962045 B RLE suglaudinti duomenys: 407230 B RLE modifikacija suglaudinti duomenys: 355826 B A&A suglaudinti duomenys: 396244 B	Eksperimentas Nr.78 Neglaudinti paveikslėlio duomenys: 227853 B RLE suglaudinti duomenys: 53256 B RLE modifikacija suglaudinti duomenys: 48486 B A&A suglaudinti duomenys: 37520 B	Eksperimentas Nr.95 Neglaudinti paveikslėlio duomenys: 562500 B RLE suglaudinti duomenys: 147486 B RLE modifikacija suglaudinti duomenys: 134444 B A&A suglaudinti duomenys: 107966 B
Eksperimentas Nr.63 Neglaudinti paveikslėlio duomenys: 944640 B RLE suglaudinti duomenys: 105922 B RLE modifikacija suglaudinti duomenys: 95968 B A&A suglaudinti duomenys: 95879 B	Eksperimentas Nr.79 Neglaudinti paveikslėlio duomenys: 442260 B RLE suglaudinti duomenys: 100834 B RLE modifikacija suglaudinti duomenys: 81732 B A&A suglaudinti duomenys: 80047 B	Eksperimentas Nr.96 Neglaudinti paveikslėlio duomenys: 630000 B RLE suglaudinti duomenys: 209930 B RLE modifikacija suglaudinti duomenys: 188236 B A&A suglaudinti duomenys: 153814 B
Eksperimentas Nr.64 Neglaudinti paveikslėlio duomenys: 988812 B RLE suglaudinti duomenys: 87450 B RLE modifikacija suglaudinti duomenys: 81512 B A&A suglaudinti duomenys: 81539 B	Eksperimentas Nr.80 Neglaudinti paveikslėlio duomenys: 282240 B RLE suglaudinti duomenys: 62046 B RLE modifikacija suglaudinti duomenys: 55610 B A&A suglaudinti duomenys: 45904 B	Eksperimentas Nr.97 Neglaudinti paveikslėlio duomenys: 218610 B RLE suglaudinti duomenys: 65880 B RLE modifikacija suglaudinti duomenys: 60126 B A&A suglaudinti duomenys: 47264 B
Eksperimentas Nr.65 Neglaudinti paveikslėlio duomenys: 780669 B RLE suglaudinti duomenys: 145822 B RLE modifikacija suglaudinti duomenys: 121368 B A&A suglaudinti duomenys: 112955 B	Eksperimentas Nr.81 Neglaudinti paveikslėlio duomenys: 493050 B RLE suglaudinti duomenys: 118502 B RLE modifikacija suglaudinti duomenys: 102890 B A&A suglaudinti duomenys: 86751 B	Eksperimentas Nr.98 Neglaudinti paveikslėlio duomenys: 745500 B RLE suglaudinti duomenys: 139906 B RLE modifikacija suglaudinti duomenys: 121192 B A&A suglaudinti duomenys: 106402 B
Eksperimentas Nr.66 Neglaudinti paveikslėlio duomenys: 430080 B RLE suglaudinti duomenys: 95146 B RLE modifikacija suglaudinti duomenys: 83374 B A&A suglaudinti duomenys: 72680 B	Eksperimentas Nr.82 Neglaudinti paveikslėlio duomenys: 134250 B RLE suglaudinti duomenys: 61684 B RLE modifikacija suglaudinti duomenys: 54712 B A&A suglaudinti duomenys: 41804 B	Eksperimentas Nr.99 Neglaudinti paveikslėlio duomenys: 204480 B RLE suglaudinti duomenys: 72148 B RLE modifikacija suglaudinti duomenys: 62860 B A&A suglaudinti duomenys: 49889 B
Eksperimentas Nr.67 Neglaudinti paveikslėlio duomenys: 450858 B RLE suglaudinti duomenys: 62138 B RLE modifikacija suglaudinti duomenys: 54020 B	Eksperimentas Nr.83 Neglaudinti paveikslėlio duomenys: 773760 B RLE suglaudinti duomenys: 244790 B RLE modifikacija suglaudinti duomenys: 213594 B A&A suglaudinti duomenys: 174508 B	Eksperimentas Nr.100 Neglaudinti paveikslėlio duomenys: 2098176 B RLE suglaudinti duomenys: 748148 B RLE modifikacija suglaudinti duomenys: 660370 B A&A suglaudinti duomenys: 525643 B
	Eksperimentas Nr.84 Neglaudinti paveikslėlio duomenys: 699033 B RLE suglaudinti duomenys: 206136 B	

6.4. priedas. Trečios dalies bandymų rezultatai

----- Eksperimentas Nr.1 Neglaudinti paveikslėlio duomenys: 147456 B RLE suglaudinti duomenys: 23780 B RLE modifikacija suglaudinti duomenys: 20064 B A&A suglaudinti duomenys: 20553 B -----	RLE modifikacija suglaudinti duomenys: 8614 B A&A suglaudinti duomenys: 9613 B -----	Neglaudinti paveikslėlio duomenys: 4536000 B RLE suglaudinti duomenys: 682438 B RLE modifikacija suglaudinti duomenys: 609270 B A&A suglaudinti duomenys: 545480 B -----
Eksperimentas Nr.2 Neglaudinti paveikslėlio duomenys: 79800 B RLE suglaudinti duomenys: 10258 B RLE modifikacija suglaudinti duomenys: 8380 B A&A suglaudinti duomenys: 9025 B -----	Eksperimentas Nr.18 Neglaudinti paveikslėlio duomenys: 5760000 B RLE suglaudinti duomenys: 1020882 B RLE modifikacija suglaudinti duomenys: 880186 B A&A suglaudinti duomenys: 804603 B -----	Eksperimentas Nr.35 Neglaudinti paveikslėlio duomenys: 1083456 B RLE suglaudinti duomenys: 149104 B RLE modifikacija suglaudinti duomenys: 132334 B A&A suglaudinti duomenys: 123747 B -----
Eksperimentas Nr.3 Neglaudinti paveikslėlio duomenys: 79800 B RLE suglaudinti duomenys: 6566 B RLE modifikacija suglaudinti duomenys: 5434 B A&A suglaudinti duomenys: 6638 B -----	Eksperimentas Nr.19 Neglaudinti paveikslėlio duomenys: 5760000 B RLE suglaudinti duomenys: 854902 B RLE modifikacija suglaudinti duomenys: 742844 B A&A suglaudinti duomenys: 703092 B -----	Eksperimentas Nr.36 Neglaudinti paveikslėlio duomenys: 1193157 B RLE suglaudinti duomenys: 117008 B RLE modifikacija suglaudinti duomenys: 104400 B A&A suglaudinti duomenys: 105997 B -----
Eksperimentas Nr.4 Neglaudinti paveikslėlio duomenys: 79800 B RLE suglaudinti duomenys: 9372 B RLE modifikacija suglaudinti duomenys: 7556 B A&A suglaudinti duomenys: 8407 B -----	Eksperimentas Nr.20 Neglaudinti paveikslėlio duomenys: 90000 B RLE suglaudinti duomenys: 19992 B RLE modifikacija suglaudinti duomenys: 16154 B A&A suglaudinti duomenys: 15406 B -----	Eksperimentas Nr.37 Neglaudinti paveikslėlio duomenys: 456750 B RLE suglaudinti duomenys: 77182 B RLE modifikacija suglaudinti duomenys: 66216 B A&A suglaudinti duomenys: 61722 B -----
Eksperimentas Nr.5 Neglaudinti paveikslėlio duomenys: 79800 B RLE suglaudinti duomenys: 18136 B RLE modifikacija suglaudinti duomenys: 15178 B A&A suglaudinti duomenys: 13612 B -----	Eksperimentas Nr.21 Neglaudinti paveikslėlio duomenys: 3932160 B RLE suglaudinti duomenys: 336952 B RLE modifikacija suglaudinti duomenys: 303730 B A&A suglaudinti duomenys: 322415 B -----	Eksperimentas Nr.38 Neglaudinti paveikslėlio duomenys: 1122000 B RLE suglaudinti duomenys: 105488 B RLE modifikacija suglaudinti duomenys: 93946 B A&A suglaudinti duomenys: 96535 B -----
Eksperimentas Nr.6 Neglaudinti paveikslėlio duomenys: 79800 B RLE suglaudinti duomenys: 4960 B RLE modifikacija suglaudinti duomenys: 4374 B A&A suglaudinti duomenys: 5166 B -----	Eksperimentas Nr.22 Neglaudinti paveikslėlio duomenys: 3686400 B RLE suglaudinti duomenys: 296014 B RLE modifikacija suglaudinti duomenys: 264348 B A&A suglaudinti duomenys: 287425 B -----	Eksperimentas Nr.39 Neglaudinti paveikslėlio duomenys: 165240 B RLE suglaudinti duomenys: 30762 B RLE modifikacija suglaudinti duomenys: 26284 B A&A suglaudinti duomenys: 23814 B -----
Eksperimentas Nr.7 Neglaudinti paveikslėlio duomenys: 79800 B RLE suglaudinti duomenys: 13112 B RLE modifikacija suglaudinti duomenys: 10636 B A&A suglaudinti duomenys: 10931 B -----	Eksperimentas Nr.23 Neglaudinti paveikslėlio duomenys: 708048 B RLE suglaudinti duomenys: 127696 B RLE modifikacija suglaudinti duomenys: 114344 B A&A suglaudinti duomenys: 101254 B -----	Eksperimentas Nr.40 Neglaudinti paveikslėlio duomenys: 1570020 B RLE suglaudinti duomenys: 243348 B RLE modifikacija suglaudinti duomenys: 217236 B A&A suglaudinti duomenys: 192894 B -----
Eksperimentas Nr.8 Neglaudinti paveikslėlio duomenys: 79800 B RLE suglaudinti duomenys: 7100 B RLE modifikacija suglaudinti duomenys: 6252 B A&A suglaudinti duomenys: 6661 B -----	Eksperimentas Nr.24 Neglaudinti paveikslėlio duomenys: 150975 B RLE suglaudinti duomenys: 28386 B RLE modifikacija suglaudinti duomenys: 25818 B A&A suglaudinti duomenys: 21906 B -----	Eksperimentas Nr.41 Neglaudinti paveikslėlio duomenys: 4800000 B RLE suglaudinti duomenys: 581674 B RLE modifikacija suglaudinti duomenys: 510282 B A&A suglaudinti duomenys: 494713 B -----
Eksperimentas Nr.9 Neglaudinti paveikslėlio duomenys: 79800 B RLE suglaudinti duomenys: 6538 B RLE modifikacija suglaudinti duomenys: 5266 B A&A suglaudinti duomenys: 6615 B -----	Eksperimentas Nr.25 Neglaudinti paveikslėlio duomenys: 4320000 B RLE suglaudinti duomenys: 287284 B RLE modifikacija suglaudinti duomenys: 262730 B A&A suglaudinti duomenys: 303051 B -----	Eksperimentas Nr.42 Neglaudinti paveikslėlio duomenys: 2359296 B RLE suglaudinti duomenys: 222900 B RLE modifikacija suglaudinti duomenys: 202402 B A&A suglaudinti duomenys: 198258 B -----
Eksperimentas Nr.10 Neglaudinti paveikslėlio duomenys: 90000 B RLE suglaudinti duomenys: 7858 B RLE modifikacija suglaudinti duomenys: 6452 B A&A suglaudinti duomenys: 7696 B -----	Eksperimentas Nr.26 Neglaudinti paveikslėlio duomenys: 405000 B RLE suglaudinti duomenys: 18718 B RLE modifikacija suglaudinti duomenys: 16312 B A&A suglaudinti duomenys: 23532 B -----	Eksperimentas Nr.43 Neglaudinti paveikslėlio duomenys: 3888000 B RLE suglaudinti duomenys: 519418 B RLE modifikacija suglaudinti duomenys: 461172 B A&A suglaudinti duomenys: 435635 B -----
Eksperimentas Nr.11 Neglaudinti paveikslėlio duomenys: 90000 B RLE suglaudinti duomenys: 6364 B RLE modifikacija suglaudinti duomenys: 5544 B A&A suglaudinti duomenys: 6433 B -----	Eksperimentas Nr.27 Neglaudinti paveikslėlio duomenys: 150876 B RLE suglaudinti duomenys: 8928 B RLE modifikacija suglaudinti duomenys: 7440 B A&A suglaudinti duomenys: 8928 B -----	Eksperimentas Nr.44 Neglaudinti paveikslėlio duomenys: 1200000 B RLE suglaudinti duomenys: 91886 B RLE modifikacija suglaudinti duomenys: 86198 B A&A suglaudinti duomenys: 88414 B -----
Eksperimentas Nr.12 Neglaudinti paveikslėlio duomenys: 90000 B RLE suglaudinti duomenys: 14250 B RLE modifikacija suglaudinti duomenys: 12070 B A&A suglaudinti duomenys: 11471 B -----	Eksperimentas Nr.28 Neglaudinti paveikslėlio duomenys: 151320 B RLE suglaudinti duomenys: 13388 B RLE modifikacija suglaudinti duomenys: 11432 B A&A suglaudinti duomenys: 12795 B -----	Eksperimentas Nr.45 Neglaudinti paveikslėlio duomenys: 3932160 B RLE suglaudinti duomenys: 599948 B RLE modifikacija suglaudinti duomenys: 528716 B A&A suglaudinti duomenys: 481716 B -----
Eksperimentas Nr.13 Neglaudinti paveikslėlio duomenys: 90000 B RLE suglaudinti duomenys: 30254 B RLE modifikacija suglaudinti duomenys: 25090 B A&A suglaudinti duomenys: 22324 B -----	Eksperimentas Nr.29 Neglaudinti paveikslėlio duomenys: 151200 B RLE suglaudinti duomenys: 30054 B RLE modifikacija suglaudinti duomenys: 26508 B A&A suglaudinti duomenys: 23314 B -----	Eksperimentas Nr.46 Neglaudinti paveikslėlio duomenys: 5760000 B RLE suglaudinti duomenys: 726654 B RLE modifikacija suglaudinti duomenys: 631384 B A&A suglaudinti duomenys: 630583 B -----
Eksperimentas Nr.14 Neglaudinti paveikslėlio duomenys: 90000 B RLE suglaudinti duomenys: 13820 B RLE modifikacija suglaudinti duomenys: 10734 B A&A suglaudinti duomenys: 11596 B -----	Eksperimentas Nr.30 Neglaudinti paveikslėlio duomenys: 692640 B RLE suglaudinti duomenys: 78334 B RLE modifikacija suglaudinti duomenys: 65538 B A&A suglaudinti duomenys: 71231 B -----	Eksperimentas Nr.47 Neglaudinti paveikslėlio duomenys: 3147264 B RLE suglaudinti duomenys: 324946 B RLE modifikacija suglaudinti duomenys: 287362 B A&A suglaudinti duomenys: 292710 B -----
Eksperimentas Nr.15 Neglaudinti paveikslėlio duomenys: 90000 B RLE suglaudinti duomenys: 17670 B RLE modifikacija suglaudinti duomenys: 14130 B A&A suglaudinti duomenys: 13906 B -----	Eksperimentas Nr.31 Neglaudinti paveikslėlio duomenys: 2508462 B RLE suglaudinti duomenys: 389416 B RLE modifikacija suglaudinti duomenys: 350432 B A&A suglaudinti duomenys: 307354 B -----	Eksperimentas Nr.48 Neglaudinti paveikslėlio duomenys: 4800000 B RLE suglaudinti duomenys: 952640 B RLE modifikacija suglaudinti duomenys: 852074 B A&A suglaudinti duomenys: 726593 B -----
Eksperimentas Nr.16 Neglaudinti paveikslėlio duomenys: 90000 B RLE suglaudinti duomenys: 17396 B RLE modifikacija suglaudinti duomenys: 15098 B A&A suglaudinti duomenys: 13121 B -----	Eksperimentas Nr.32 Neglaudinti paveikslėlio duomenys: 2477466 B RLE suglaudinti duomenys: 336096 B RLE modifikacija suglaudinti duomenys: 302644 B A&A suglaudinti duomenys: 274830 B -----	Eksperimentas Nr.49 Neglaudinti paveikslėlio duomenys: 2088960 B RLE suglaudinti duomenys: 221228 B RLE modifikacija suglaudinti duomenys: 200880 B A&A suglaudinti duomenys: 192906 B -----
Eksperimentas Nr.17 Neglaudinti paveikslėlio duomenys: 90000 B RLE suglaudinti duomenys: 10634 B -----	Eksperimentas Nr.33 Neglaudinti paveikslėlio duomenys: 2477466 B RLE suglaudinti duomenys: 349128 B RLE modifikacija suglaudinti duomenys: 316158 B A&A suglaudinti duomenys: 282422 B -----	Eksperimentas Nr.50 Neglaudinti paveikslėlio duomenys: 2359296 B RLE suglaudinti duomenys: 253986 B RLE modifikacija suglaudinti duomenys: 225456 B A&A suglaudinti duomenys: 224955 B -----
-----	Eksperimentas Nr.34 -----	-----

<p>----- Eksperiments Nr.51 Neglaudinti paveikslēlio duomenys: 1157238 B RLE suglaudinti duomenys: 131532 B RLE modifikacija suglaudinti duomenys: 110318 B A&A suglaudinti duomenys: 115031 B -----</p>	<p>RLE modifikacija suglaudinti duomenys: 76660 B A&A suglaudinti duomenys: 74000 B -----</p>	<p>Neglaudinti paveikslēlio duomenys: 1440000 B RLE suglaudinti duomenys: 256270 B RLE modifikacija suglaudinti duomenys: 223228 B A&A suglaudinti duomenys: 201053 B -----</p>
<p>Eksperiments Nr.52 Neglaudinti paveikslēlio duomenys: 2073600 B RLE suglaudinti duomenys: 423432 B RLE modifikacija suglaudinti duomenys: 368122 B A&A suglaudinti duomenys: 325327 B -----</p>	<p>Eksperiments Nr.69 Neglaudinti paveikslēlio duomenys: 721800 B RLE suglaudinti duomenys: 92192 B RLE modifikacija suglaudinti duomenys: 76360 B A&A suglaudinti duomenys: 81565 B -----</p>	<p>Eksperiments Nr.87 Neglaudinti paveikslēlio duomenys: 968100 B RLE suglaudinti duomenys: 156086 B RLE modifikacija suglaudinti duomenys: 137538 B A&A suglaudinti duomenys: 126416 B -----</p>
<p>Eksperiments Nr.53 Neglaudinti paveikslēlio duomenys: 718200 B RLE suglaudinti duomenys: 72136 B RLE modifikacija suglaudinti duomenys: 66308 B A&A suglaudinti duomenys: 63091 B -----</p>	<p>Eksperiments Nr.70 Neglaudinti paveikslēlio duomenys: 524724 B RLE suglaudinti duomenys: 65768 B RLE modifikacija suglaudinti duomenys: 56930 B A&A suglaudinti duomenys: 56558 B -----</p>	<p>Eksperiments Nr.88 Neglaudinti paveikslēlio duomenys: 608652 B RLE suglaudinti duomenys: 135212 B RLE modifikacija suglaudinti duomenys: 119894 B A&A suglaudinti duomenys: 100081 B -----</p>
<p>Eksperiments Nr.54 Neglaudinti paveikslēlio duomenys: 1005600 B RLE suglaudinti duomenys: 110736 B RLE modifikacija suglaudinti duomenys: 96770 B A&A suglaudinti duomenys: 95492 B -----</p>	<p>Eksperiments Nr.71 Neglaudinti paveikslēlio duomenys: 499500 B RLE suglaudinti duomenys: 52870 B RLE modifikacija suglaudinti duomenys: 46068 B A&A suglaudinti duomenys: 48498 B -----</p>	<p>Eksperiments Nr.89 Neglaudinti paveikslēlio duomenys: 1166400 B RLE suglaudinti duomenys: 178646 B RLE modifikacija suglaudinti duomenys: 151978 B A&A suglaudinti duomenys: 148791 B -----</p>
<p>Eksperiments Nr.55 Neglaudinti paveikslēlio duomenys: 1966080 B RLE suglaudinti duomenys: 220280 B RLE modifikacija suglaudinti duomenys: 195594 B A&A suglaudinti duomenys: 189057 B -----</p>	<p>Eksperiments Nr.72 Neglaudinti paveikslēlio duomenys: 716943 B RLE suglaudinti duomenys: 44232 B RLE modifikacija suglaudinti duomenys: 40974 B A&A suglaudinti duomenys: 47375 B -----</p>	<p>Eksperiments Nr.90 Neglaudinti paveikslēlio duomenys: 583200 B RLE suglaudinti duomenys: 104492 B RLE modifikacija suglaudinti duomenys: 90568 B A&A suglaudinti duomenys: 81059 B -----</p>
<p>Eksperiments Nr.56 Neglaudinti paveikslēlio duomenys: 790020 B RLE suglaudinti duomenys: 142392 B RLE modifikacija suglaudinti duomenys: 124552 B A&A suglaudinti duomenys: 111399 B -----</p>	<p>Eksperiments Nr.73 Neglaudinti paveikslēlio duomenys: 275520 B RLE suglaudinti duomenys: 35054 B RLE modifikacija suglaudinti duomenys: 30318 B A&A suglaudinti duomenys: 30402 B -----</p>	<p>Eksperiments Nr.91 Neglaudinti paveikslēlio duomenys: 633360 B RLE suglaudinti duomenys: 80668 B RLE modifikacija suglaudinti duomenys: 71126 B A&A suglaudinti duomenys: 69094 B -----</p>
<p>Eksperiments Nr.57 Neglaudinti paveikslēlio duomenys: 2359296 B RLE suglaudinti duomenys: 327824 B RLE modifikacija suglaudinti duomenys: 290468 B A&A suglaudinti duomenys: 269797 B -----</p>	<p>Eksperiments Nr.74 Neglaudinti paveikslēlio duomenys: 443625 B RLE suglaudinti duomenys: 34590 B RLE modifikacija suglaudinti duomenys: 31784 B A&A suglaudinti duomenys: 33983 B -----</p>	<p>Eksperiments Nr.92 Neglaudinti paveikslēlio duomenys: 402300 B RLE suglaudinti duomenys: 83902 B RLE modifikacija suglaudinti duomenys: 72414 B A&A suglaudinti duomenys: 63254 B -----</p>
<p>Eksperiments Nr.58 Neglaudinti paveikslēlio duomenys: 562500 B RLE suglaudinti duomenys: 87152 B RLE modifikacija suglaudinti duomenys: 74876 B A&A suglaudinti duomenys: 70209 B -----</p>	<p>Eksperiments Nr.75 Neglaudinti paveikslēlio duomenys: 1531200 B RLE suglaudinti duomenys: 158392 B RLE modifikacija suglaudinti duomenys: 140322 B A&A suglaudinti duomenys: 144839 B -----</p>	<p>Eksperiments Nr.93 Neglaudinti paveikslēlio duomenys: 568980 B RLE suglaudinti duomenys: 91110 B RLE modifikacija suglaudinti duomenys: 81230 B A&A suglaudinti duomenys: 72781 B -----</p>
<p>Eksperiments Nr.59 Neglaudinti paveikslēlio duomenys: 6912000 B RLE suglaudinti duomenys: 375350 B RLE modifikacija suglaudinti duomenys: 355598 B A&A suglaudinti duomenys: 413709 B -----</p>	<p>Eksperiments Nr.76 Neglaudinti paveikslēlio duomenys: 374850 B RLE suglaudinti duomenys: 31862 B RLE modifikacija suglaudinti duomenys: 28428 B A&A suglaudinti duomenys: 30350 B -----</p>	<p>Eksperiments Nr.94 Neglaudinti paveikslēlio duomenys: 593616 B RLE suglaudinti duomenys: 98446 B RLE modifikacija suglaudinti duomenys: 85624 B A&A suglaudinti duomenys: 76840 B -----</p>
<p>Eksperiments Nr.60 Neglaudinti paveikslēlio duomenys: 1822500 B RLE suglaudinti duomenys: 214652 B RLE modifikacija suglaudinti duomenys: 189940 B A&A suglaudinti duomenys: 184612 B -----</p>	<p>Eksperiments Nr.77 Neglaudinti paveikslēlio duomenys: 243000 B RLE suglaudinti duomenys: 48284 B RLE modifikacija suglaudinti duomenys: 40738 B A&A suglaudinti duomenys: 37719 B -----</p>	<p>Eksperiments Nr.95 Neglaudinti paveikslēlio duomenys: 562500 B RLE suglaudinti duomenys: 71588 B RLE modifikacija suglaudinti duomenys: 63060 B A&A suglaudinti duomenys: 60558 B -----</p>
<p>Eksperiments Nr.61 Neglaudinti paveikslēlio duomenys: 867825 B RLE suglaudinti duomenys: 117706 B RLE modifikacija suglaudinti duomenys: 100326 B A&A suglaudinti duomenys: 99582 B -----</p>	<p>Eksperiments Nr.78 Neglaudinti paveikslēlio duomenys: 227853 B RLE suglaudinti duomenys: 35322 B RLE modifikacija suglaudinti duomenys: 30862 B A&A suglaudinti duomenys: 28848 B -----</p>	<p>Eksperiments Nr.96 Neglaudinti paveikslēlio duomenys: 630000 B RLE suglaudinti duomenys: 93502 B RLE modifikacija suglaudinti duomenys: 79306 B A&A suglaudinti duomenys: 78172 B -----</p>
<p>Eksperiments Nr.62 Neglaudinti paveikslēlio duomenys: 4962045 B RLE suglaudinti duomenys: 312170 B RLE modifikacija suglaudinti duomenys: 279614 B A&A suglaudinti duomenys: 336425 B -----</p>	<p>Eksperiments Nr.79 Neglaudinti paveikslēlio duomenys: 442260 B RLE suglaudinti duomenys: 69136 B RLE modifikacija suglaudinti duomenys: 56548 B A&A suglaudinti duomenys: 56689 B -----</p>	<p>Eksperiments Nr.97 Neglaudinti paveikslēlio duomenys: 218610 B RLE suglaudinti duomenys: 32836 B RLE modifikacija suglaudinti duomenys: 28368 B A&A suglaudinti duomenys: 26396 B -----</p>
<p>Eksperiments Nr.63 Neglaudinti paveikslēlio duomenys: 944640 B RLE suglaudinti duomenys: 60842 B RLE modifikacija suglaudinti duomenys: 56846 B A&A suglaudinti duomenys: 64911 B -----</p>	<p>Eksperiments Nr.80 Neglaudinti paveikslēlio duomenys: 282240 B RLE suglaudinti duomenys: 44004 B RLE modifikacija suglaudinti duomenys: 37904 B A&A suglaudinti duomenys: 36449 B -----</p>	<p>Eksperiments Nr.98 Neglaudinti paveikslēlio duomenys: 745500 B RLE suglaudinti duomenys: 81370 B RLE modifikacija suglaudinti duomenys: 70618 B A&A suglaudinti duomenys: 71014 B -----</p>
<p>Eksperiments Nr.64 Neglaudinti paveikslēlio duomenys: 988812 B RLE suglaudinti duomenys: 63146 B RLE modifikacija suglaudinti duomenys: 59220 B A&A suglaudinti duomenys: 66639 B -----</p>	<p>Eksperiments Nr.81 Neglaudinti paveikslēlio duomenys: 493050 B RLE suglaudinti duomenys: 60156 B RLE modifikacija suglaudinti duomenys: 52242 B A&A suglaudinti duomenys: 50505 B -----</p>	<p>Eksperiments Nr.99 Neglaudinti paveikslēlio duomenys: 204480 B RLE suglaudinti duomenys: 40234 B RLE modifikacija suglaudinti duomenys: 34482 B A&A suglaudinti duomenys: 30482 B -----</p>
<p>Eksperiments Nr.65 Neglaudinti paveikslēlio duomenys: 780669 B RLE suglaudinti duomenys: 105056 B RLE modifikacija suglaudinti duomenys: 87738 B A&A suglaudinti duomenys: 88167 B -----</p>	<p>Eksperiments Nr.82 Neglaudinti paveikslēlio duomenys: 134250 B RLE suglaudinti duomenys: 28694 B RLE modifikacija suglaudinti duomenys: 24610 B A&A suglaudinti duomenys: 21761 B -----</p>	<p>Eksperiments Nr.100 Neglaudinti paveikslēlio duomenys: 2098176 B RLE suglaudinti duomenys: 369662 B RLE modifikacija suglaudinti duomenys: 322436 B A&A suglaudinti duomenys: 289715 B -----</p>
<p>Eksperiments Nr.66 Neglaudinti paveikslēlio duomenys: 430080 B RLE suglaudinti duomenys: 61998 B RLE modifikacija suglaudinti duomenys: 52966 B A&A suglaudinti duomenys: 52409 B -----</p>	<p>Eksperiments Nr.83 Neglaudinti paveikslēlio duomenys: 773760 B RLE suglaudinti duomenys: 137218 B RLE modifikacija suglaudinti duomenys: 116696 B A&A suglaudinti duomenys: 109527 B -----</p>	
<p>Eksperiments Nr.67 Neglaudinti paveikslēlio duomenys: 450858 B RLE suglaudinti duomenys: 50066 B RLE modifikacija suglaudinti duomenys: 42378 B A&A suglaudinti duomenys: 44803 B -----</p>	<p>Eksperiments Nr.84 Neglaudinti paveikslēlio duomenys: 699033 B RLE suglaudinti duomenys: 117756 B RLE modifikacija suglaudinti duomenys: 100390 B A&A suglaudinti duomenys: 92608 B -----</p>	
<p>Eksperiments Nr.68 Neglaudinti paveikslēlio duomenys: 469404 B RLE suglaudinti duomenys: 94098 B -----</p>	<p>Eksperiments Nr.85 Neglaudinti paveikslēlio duomenys: 562500 B RLE suglaudinti duomenys: 128110 B RLE modifikacija suglaudinti duomenys: 111412 B A&A suglaudinti duomenys: 95632 B -----</p>	
	<p>Eksperiments Nr.86</p>	