

Андасова Б.З., Ташатов Н.Н., Сатыбалдина Д.Ж.

АҚПАРАТТЫ КОДТАУ

оқу құралы

65644



ЭВЕРО
Алматы, 2022

УДК 004.4(075)
ББК 32.973.202.я73
А57

Л.Н. Гумилев атындағы Еуразия ұлттық университеті Ғылыми кеңесі ұсынған
2017 жылғы 29 маусымдағы №14 хаттамасы.

Пікір жазғандар:

Маткаримов Б.Ж., техника ғылымдарының докторы, профессор

Мукашева М.У., педагогика ғылымдарының кандидаты, доцент

Тусупов А.А., физика-математика ғылымдарының докторы, профессор

А57 **Андасова Б.З., Ташатов Н.Н., Сатыбалдина Д.Ж.**

Ақпаратты кодтау: ақпараттық технология мамандықтарына арналған оқу құралы. – Алматы: Эверо, 2022. -132 б.

ISBN 978-601-327-252-8

Оқу құралында кодтау теориясының негізгі ұғымдары мен жіктелуі, сондай-ақ ақпараттың шығу көздерін кодтау, деректерді сығу үшін қолданылатын кодтау әдістері мен алгоритмдері қарастырылады. Әр бөлім тақырыбы бойынша мысалдар, бақылау сұрақтары, өздік жұмыс тапсырмалары беріледі.

Оқу құралы ақпараттық технологиялар мамандықтарының білім алушыларына және оқытушыларға арналған.

УДК 004.4(075)
ББК 32.973.202.я73

ISBN 978-601-327-252-8

© **Андасова Б.З.**

Ташатов Н.Н.

Сатыбалдина Д.Ж., 2022

© Эверо, 2022

МАЗМУНЫ

| | |
|--|-----|
| Алғысөз | 5 |
| Кіріспе | 6 |
| 1 КОДТАУ ТЕОРИЯСЫНЫҢ НЕГІЗГІ ҰҒЫМДАРЫ | 7 |
| 1.1 Ақпаратты кодтау және код сипаттамалары | 7 |
| 1.1.1 Код және кодтау ұғымдары | 7 |
| 1.1.2 Кодтардың мысалдары | 10 |
| 1.1.3 Цифрлық байланыс арнасында ақпараттың жіберілуі. | 15 |
| 1.1.4 Кодтың негізгі сипаттамалары | 18 |
| 1.1.5 Кодты сипаттайтын негізгі ұғымдар | 18 |
| 1.1.6 Кодтарды бейнелеу тәсілдері | 21 |
| 1.2 Қодтау теориясының даму тарихы | 22 |
| 1.3 Екілік кодтардың жіктелуі | 31 |
| 1.4 Екілік кодтардың түрлері | 35 |
| 1.4.1 Артықтығы бар кодтар | 35 |
| 1.4.2 Артықтығы жоқ кодтар | 42 |
| 2 АҚПАРАТТЫҢ ШЫҒУ КӨЗДЕРІН КОДТАУ | 54 |
| 2.1 Ақпараттың шығу көздерін кодтау теоремалары | 54 |
| 2.1.1 Шеннонның 1-ші теоремасы | 54 |
| 2.1.2 Шеннонның 2-ші теоремасы | 54 |
| 2.2 Тиімді кодтар | 57 |
| 2.3 Шеннон-Фано коды | 57 |
| 2.4 Хаффман коды | 59 |
| 3 ДЕРЕКТЕРДІ СЫҒУ | 67 |
| 3.1 Деректерді сығу ұғымы | 67 |
| 3.2 Сығу түрлері, әдістері, алгоритмдерінің жіктелуі | 70 |
| 3.2.1 Сығу түрлері | 70 |
| 3.2.2 Деректерді сығу әдістері мен құралдары | 75 |
| 3.2.3 Сығу алгоритмдерінің жіктелуі | 77 |
| 3.3 Арифметикалық кодтау | 78 |
| 3.4 Лемпель-Зив кодтары | 87 |
| 3.4.1 LZ77 (Лемпель-Зив) алгоритмі | 87 |
| 3.4.2 LZ78 (Лемпель-Зив) алгоритмі | 90 |
| 3.4.3 LZSS (Лемпель-Зив-Сторер-Шиманский) алгоритмі | 92 |
| 3.4.4 LZW (Лемпель-Зив-Уелч) алгоритмі | 94 |
| 3.5 Графикалық ақпаратты сығу | 100 |
| 3.5.1 Кескіндерді сығу | 100 |

| | |
|--|------------|
| 3.5.2 Бейне ақпаратты сығу | 103 |
| 3.5.3 Графиктік ақпаратты сығу алгоритмдері, технологиялары | 104 |
| 3.6 Дыбыстық ақпаратты сығу әдістері | 120 |
| Қорытынды | 128 |
| Пайдаланылған әдебиеттер тізімі | 130 |

АЛҒЫ СӨЗ

Кодтау теориясы бойынша қазақ тіліндегі оқу басылымдарына деген сұранысты ескере отырып жазылған бұл оқу құралы екілік кодтау теориясы туралы негізгі ұғымдарды, екілік кодтар арқылы ақпаратты кодтау мен декодтау әдістерін таныстыруға арналған.

Оқу құралында қазіргі таңда ақпараттық технологиялар саласының мамандары білуі тиіс ақпаратты кодтау мен арнаны кодтауға қатысты негізгі ұғымдар, код сипаттамалары, екілік кодтар мен бөгеуілге орнықты кодтардың жіктелуі, тиімді кодтау, деректерді сығуда қолданылатын кодтау алгоритмдері және кодтау мен декодтау әдістері сипатталады. Қарастырылған негізгі ұғымдар нақты мысалдармен, кестелермен түсіндіріледі, бөлімдер соңында тақырыпқа сәйкес бақылау сұрақтары және өзіндік жұмыс тапсырмалары беріледі.

Оқу құралы авторлардың Л.Н.Гумилев атындағы Еуразия ұлттық университетінің ақпараттық технологиялар факультетінің студенттеріне оқыған «Кодтау теориясы» атты таңдау курсының дәрістік материалдары негізінде даярланған, аталған пәннің жұмыстық оқу бағдарламасындағы көрсетілген тақырыптарға толық сәйкес келеді.

Оқу құралы кодтау теориясына қатысты пәнді оқитын ақпараттық технологиялар саласының 5В060200-«Информатика», 5В011100-«Информатика», 5В070400-«Есептеу техникасы және бағдарламалық қамтамасыз ету», 5В070300-«Ақпараттық жүйелер», 5В070200-«Автоматтандыру және басқару», 5В100200-«Ақпараттық қауіпсіздік жүйелері» мамандықтарының студенттеріне арналған. Оқу құралының материалдарын «Деректерді қорғау және ақпараттық қауіпсіздік», «Бөгеуілге орнықты кодтау», «Ақпараттық жүйелердің негіздері» сияқты пәндерді де оқыту үшін, осы кодтау теориясы бойынша негізгі ұғымдарды білуді қажет ететін криптография, компьютерлік желілер, ақпараттық қауіпсіздік, автоматтандырылған ақпараттық жүйелерді құруға байланысты пәндерді оқытуда дәріске қосымша материал ретінде, сондай-ақ студенттердің өздік жұмыстарында, емтиханға дайындықта, курстық және диплом жұмыстарын орындағанда қолдануға болады.

КІРІСПЕ

Қазіргі уақытта кодтау теориясы ғылымның маңызды саласының біріне айналып отыр және арнамен берілетін ақпаратты өңдеу мен жіберуді жүзеге асыру кодтау әдістерінсіз мүмкін емес. Сондықтан да, ақпараттық технологиялар саласының мамандықтарының студенттеріне осы пәнді оқыту үшін оны әдістемелік жағынан қамтамасыз ету өзекті болып табылады.

Сондықтан да, оқу құралында цифрлық байланыс арналарында қолданылатын екілік кодтардың түрлері сипатталып, ақпарат көзін және арнадағы хабарды кодтау мен декодтау әдістері көрсетіледі. Бұл студенттердің екілік кодтау теориясының негіздерімен, бөгеуілге орнықты кодтау және декодтау әдістерімен танысуына жағдай жасап, олардың өздігінен білім алу мүмкіндігін арттырады.

Оқу құралының құрылымы 3 бөлімнен тұрады. Бөлімдер пәннің оқу бағдарламасының тақырыптарына сәйкес, тақырыптардың бірізділігі мен сабақтастығын, логикалық байланысын сақтап орналасқан. Оқу құралында екілік кодтау теориясының негізгі кодтарына шолу жасалып, бөгеуілге орнықты кодтау әдістеріне арналған бөлімдерге аса мән беріледі.

«Кодтау теориясының негізгі ұғымдары» атты бірінші бөлімде кодтау теориясының негізгі ұғымдары мен тарихы, кодтың негізгі сипаттамалары, екілік кодтардың жіктелуі мен түрлері сипатталады.

«Ақпараттың шығу көздерін кодтау» атты екінші бөлімде ақпараттың шығу көздерін кодтаудың теоремалары, тиімді кодтау ұғымы, тиімді кодтардың түрлері, олардың кодтау және декодтау алгоритмдері қарастырылады.

«Деректерді сығу» атты үшінші бөлімде деректерді сығу ұғымы, сығу алгоритмдерінің жіктелуі, арифметикалық кодтау мен Лемпель-Зив кодтарының алгоритмдері сипатталады. Графикалық және дыбыстық ақпараттарды сығу әдістері туралы ақпарат беріледі.

Оқу құралында кітап соңында көрсетілген әдебиеттердегі материалдар, келтірілген мысалдар пайдаланылды.

Оқу құралының мазмұны кітаптың басында, ал қорытынды және пайдаланылған әдебиеттер тізімі соңында беріледі.

1 ЕКІЛІК КОДТАУ ТЕОРИЯСЫНЫҢ НЕГІЗГІ ҰҒЫМДАРЫ

1.1 Ақпаратты кодтау және код сипаттамалары

1.1.1 Код және кодтау ұғымдары

Арнаға жіберілетін ақпараттың жоғары шынайылығы мен оның сенімді сақталуын қамтамасыз ету тұрғысынан деректерді жіберумен, сақтаумен және өңдеумен байланысты есептер есептеу жүйелерінің мүмкіндіктеріне жоғары талап қойылады. Цифрлық деректерді сақтау мен өңдеу шынайылығын арттыруда кодтау теориясының үлесі зор. Жіберілген ақпарат байланыс арнасында түрлі шулар, бөгеуілдер әсерінен өзгеріске ұшырап, бұрмалануы мүмкін. Сондықтан да қазіргі заманғы бөгеуілге орнықты кодтау әдістері цифрлық байланыс арнасынан қабылданған хабардағы қателерді анықтап, түзетуге, бастапқы хабарды қалпына келтіруге мүмкіндік береді.

Ескерту. Кодтау теориясының даму тарихы, бөгеуілге орнықты кодтаудың зерттеу бағыттары туралы мына әдебиеттен [8, Королёв А.И. Коды и устройства помехоустойчивого кодирования] толығырақ білуге болады.

Кодтау теориясының негізгі міндеті байланыс арнасымен жіберілетін ақпараттың белгіленген дәлдігін және хабардың құнын барынша арзан, әрі мүмкіндігінше жылдам берілуін қамтамасыз ететін кодтарды құрастыру үшін ақпарат көздерінің сипаттамалары мен байланыс арналарын оңтайлы пайдалану болып табылады.

Ақпарат және кодтау – қазіргі заманғы ақпараттық технологияның басты ұғымдары. Оқырманды ақпарат ұғымымен, ақпарат теориясының ұғымдарымен таныс деп санап, оларға тоқталмай, кодтауға қатысты негізгі ұғымдармен таныстырайық.

Ескерту. Ақпарат теориясына қатысты материалдармен кітап соңында берілген [1], [3], [5], [10], [23] әдебиеттерден, ал цифрлық байланыс арнасының сипаттамалары, яғни сигналдар мен бөгеуілдер, ақпарат көздерін дискреттеу, кванттау, сигналдарды аналогтік-цифрлық түрлендірулер және ақпарат теориясының кодтаумен байланысты ұғымдары туралы [13], [19], [22], [24] және басқа әдебиеттерден танысуға болады.

Ақпарат байланыс арнасы арқылы жіберілгенде арнаға хабар түрінде келіп түседі.

Хабар дегеніміз ақпаратты құрайтын символдар мен алғашқы сигналдар жиынтығы. Әр түрлі хабар мысалдарына ЭЕМ-нен алынған цифрлық деректер, сөйленген сөздер, телеграмма мәтіндері, басқару командалары, әр түрлі физикалық шамаларды өлшеу нәтижелері және т.б. арнамен жіберілетін ақпарат түрлері жатады. Бұл хабар түрлерінің барлығы алдын-ала бастапқы хабардың барлық қасиеттерін сақтайтын сигналдарға түрлендіріледі, содан кейін олар әрі қарай байланыс желісімен жіберуге болатындай түрде беріледі.

Байланыс арнасы – бұл хабарды жіберуге арналған құралдар жиынтығы.

Байланыс желісі – бұл жіберушіден қабылдағышқа сигналдардың келіп түсуін қамтамасыз ететін физикалық орта (ауа, металл, магниттік таспа және т.б.).

Хабар *дискреттік және үздіксіз* (үзіліссіз) болып екіге бөлінеді.

Дискреттік хабарлар ақпарат көзі арқылы жеке элементтерді, яғни символдарды тізбектеу нәтижесінде құрылады. Хабарды құрайтын түрлі символдар жиынтығын *хабар көзінің әліпбиі* (алфавит), ал әліпбидегі символдар санын *әліпби қуаты* (көлемі) деп атайды. Символдар ретінде табиғи немесе жасанды тілдің әріптері болуы мүмкін.

Үздіксіз хабарлар элементтерге бөлінбейді, олар мәндердің үздіксіз жиынтығын қабылдайтын уақыт функцияларымен сипатталады. Үздіксіз хабардың мысалына сөйлеу (речь) және теледидарлық бейне жатады. Байланыс жүйесінде ақпаратты жіберу сапасын арттыру үшін үздіксіз хабарларды дискреттік хабарларға түрлендіреді.

Кодтау теориясының негізгі міндеті жіберілетін ақпараттың белгіленген дәлдігін және хабардың құнын барынша арзан болатындай етіп, әрі мүмкіндігінше оның арнамен жылдам берілуін қамтамасыз ететін кодтарды құрастыру үшін хабар көздерінің ақпараттық сипаттамасы мен байланыс арналарын оңтайлы пайдалану болып табылады.

Ақпаратты кодтау деп арнамен жіберілетін хабарды белгілі бір сигналдардың тізбегіне түрлендіру операциясын айтады. Ал қабылданған сигналдар бойынша хабардың мазмұнын бастапқы қалпына қайта келтіретін кері операцияны (процесті) **декодтау** деп атайды.

Код дегеніміз ақпаратты сақтағанда, арнамен жібергенде және өңдегенде әрбір нақты бастапқы хабарға қатаң түрде белгіленген әртүрлі символдар комбинациясын сәйкес келтіру тәртібі (алгоритмі, ережесі).

Кодтау деп бастапқы хабарды белгілі ереже бойынша әртүрлі символдар комбинациясына (немесе оларға сәйкес келетін дискреттік сигналдар комбинациясына) түрлендіру процесін айтады. Кодтау арқылы бір әліпби символдарын басқа әліпби символдарына бірмәнді түрде түрлендіруге болады. Осы тұрғыдан *код* дегеніміз осы түрлендірулерді жүзеге асыратын тәртіп (ереже, алгоритм) болады.

Әрбір комбинация қандай да бір символдардан, яғни кодтық комбинация элементтерінен құралған тізбек түрінде жазылады. Кодтық комбинация элементтері ретінде символдар, әріптер (А, В, С...) және сандар (0, 1, 2...) қолданылады.

Техникалық ақпараттық құрылғыларда элемент ретінде тұрақты токтың (видеоимпульс) және айнымалы токтың (радиоимпульс) токтың бірлік импульстары, олардың арасындағы үзіліс қарастырылады. Бұл элементтер қандай да бір немесе бірнеше кодтық (импульстік) белгілерге байланысты ажыратылады. Кодтық белгілерге мәні, полярлық, уақыт, фаза, жиілік сияқты параметрлер жатады.

Әрбір хабар үшін анықталған кодтық комбинация оған бірмәнді түрде сәйкес келуі тиіс, яғни кодталған хабарды декодтағанда сол бастапқы хабар алынуы тиіс. Код барлық хабарды берілген хабардың жиынтығына ортақ болатын қандай да бір тілде жазуға мүмкіндік береді. Бұл тұрғыдан берілген кодтың элементтер жиынтығы әліпби ретінде, ал осы элементтердің кодтық комбинациялары *кодтық сөз* ретінде қарастырылады. Әрбір хабарға сәйкес келетін кодтық сөз арнаға жіберіледі.

Хабарды осылай кодтық комбинацияға (кодтық сөзге) түрлендіру келесі мүмкіндіктерді береді:

– берілген байланыс арнасы арқылы n элементтен тұратын (комбинацияланған), m кодтық белгілері бар әр түрлі хабардың қажетті санын жіберуге;

– байланыс арнасының параметрлері мен жіберілген хабар арасындағы өзара келісушіліктің, үйлесімнің болуын;

- жіберілетін хабардың шынайылығын арттыруға, яғни хабар мазмұнының бұрмаланбай дұрыс болуына;
- байланыс арнасын, оның жолақтарын тиімді пайдалануға;
- хабарды жіберу мен сақтау құнын (жететін шығындарды) азайтуға;
- ақпаратты қорғау мақсатында жіберілетін хабарды жасыруға.

Осы аталған мүмкіндіктердің барлығының орындалуын қамтитын кодтау әдісін таңдау күрделі болып табылады. Өйткені, ол мына факторларға: жіберілетін хабар санына, кодтық белгілер санына, жіберуге қажетті уақытқа, байланыс арнасының параметрлеріне, аппаратуралық іске асыру мүмкіндігіне тәуелді болады.

Жалпы жағдайда кодтау әдісінің сапасы жіберудің тең жылдамдығында талап етілген бөгеуілге орнықтылыққа жету үшін сигнал көлемімен бағаланады.

Егер әліпбиде 2 элемент (әріп) болса, мысалы, 0 және 1, мұндай код (әліпби) *екілік код* немесе или бинарлық код деп аталады. Егер қарапайым сигналдар (әріптер, символдар) саны 2-ден артық болса, онда мұндай кодты *көпмәнді код* (әліпби) деп атайды.

Мысалы, әліпбидегі символдар саны он (0, 1, 2, 3, 4, 5, 6, 7, 8, 9) болатын код (әліпби) *ондық код* деп аталады.

1.1.2 Кодтардың мысалдары

Кодтарға бірнеше мысалдар келтірейік.

Халықаралық телеграф коды (МТК1, МТК2). Бұл код туралы [1] әдебиетті қараңыз.

Екілік код: 0,1,10,11,100,101,110,111,1000

Он алтылық код: 0,1,2...9, А, В, С, D, E, F.

Unicode коды. Unicode кодтауымен 65535 символды сипаттауға болады, әр символ екі байттық кодпен бейнеленеді. Бастапқы 128 символдар ASCII кодының символдарымен сәйкес келеді. Ұлттық әліпбидің символдарын «\» тізбегі арқылы кодтауға болады. Одан кейін UTF-16 немесе UTF-8 кодтық кестелері үшін символдың он алтылық кодтау жүйесіндегі коды жазылады. Unicode кодтауының барлық символдарын \u0000 пен \uFFFF аралығында кодтауға болады.

Мысалы: \u0030..... \u0039 – бұл ISO-LATIN кодтауындағы 0-9 аралығындағы цифрлар. Ал \u0024 – бұл \$ (доллар) белгісі. Unicode кодтауындағы қазақ әліпбиінің әріптері '0410' –'04D0' аралығында орналасқан (<http://unicode-table.com/ru/> сайтың қараңыз).

Орыс әліпбиінің әріптері ' \u0410 ' – А бас әрпі мен ' \u0422 ' – Я өле әрпі аралығында, ал кіші әріптер ' \u0430 ' – а әрпі мен ' \u0442 ' – я әрпі аралығында орналасады. Тек Ё әрпінің коды \u0401 және ё әрпінің коды \u0451 болады.

КОИ-8 коды – қарапайым код, компьютерде орыс әріптерін кодтаудың алғашқы стандарты (Код Обмена Информацией – ақпарат алмасу коды, 8-биттік).

| | | | | | | | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 |
| 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 |
| 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 |
| 80 | 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 | 91 | 92 | 93 | 94 | 95 |
| 96 | 97 | 98 | 99 | 100 | 101 | 102 | 103 | 104 | 105 | 106 | 107 | 108 | 109 | 110 | 111 |
| 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 | 120 | 121 | 122 | 123 | 124 | 125 | 126 | 127 |
| 128 | 129 | 130 | 131 | 132 | 133 | 134 | 135 | 136 | 137 | 138 | 139 | 140 | 141 | 142 | 143 |
| 144 | 145 | 146 | 147 | 148 | 149 | 150 | 151 | 152 | 153 | 154 | 155 | 156 | 157 | 158 | 159 |
| 160 | 161 | 162 | 163 | 164 | 165 | 166 | 167 | 168 | 169 | 170 | 171 | 172 | 173 | 174 | 175 |
| 176 | 177 | 178 | 179 | 180 | 181 | 182 | 183 | 184 | 185 | 186 | 187 | 188 | 189 | 190 | 191 |
| 192 | 193 | 194 | 195 | 196 | 197 | 198 | 199 | 200 | 201 | 202 | 203 | 204 | 205 | 206 | 207 |
| 208 | 209 | 210 | 211 | 212 | 213 | 214 | 215 | 216 | 217 | 218 | 219 | 220 | 221 | 222 | 223 |
| 224 | 225 | 226 | 227 | 228 | 229 | 230 | 231 | 232 | 233 | 234 | 235 | 236 | 237 | 238 | 239 |
| 240 | 241 | 242 | 243 | 244 | 245 | 246 | 247 | 248 | 249 | 250 | 251 | 252 | 253 | 254 | 255 |

№14 МИКРОТІШІМШІ ОПАВК.

Трисиме коды – теңқалыпты код, латын әліпбиінің символдарына 1,2,3 символдарының комбинациялары сәйкес келеді.

| | | | | | |
|---|-----|---|-----|---|-----|
| A | 111 | J | 211 | S | 311 |
| B | 112 | K | 212 | T | 312 |
| C | 113 | L | 213 | U | 313 |
| D | 121 | M | 221 | V | 321 |
| E | 122 | N | 222 | W | 322 |
| F | 123 | O | 223 | X | 323 |
| G | 131 | P | 231 | Y | 331 |
| H | 132 | Q | 232 | Z | 332 |
| I | 133 | R | 233 | Z | 333 |

Морзе коды – тең калыпты емес телеграфтық код. Морзе әліпшесі америкалық өнертапқыш Сэмюэл Морзе (1791 –1872) ойлап тапқан аппарат арқылы жүзеге асырылады. Бұл Морзе аппараты хабарларды кодпен беруге және қағаз таспаға кодпен жазып алуға мүмкіндік береді.

| Орыс әрпі | Латын әрпі | Морзе коды | Цифр | Морзе коды | Символ | Морзе коды |
|-----------|------------|------------|------|------------|--------|----------------|
| А | A | ••••• | 0 | ••••• | . | •••••••• |
| Б | B | ••••• | 1 | ••••• | , | •••••••••• |
| В | W | ••••• | 2 | ••••• | : | •••••••••••• |
| Г | G | ••••• | 3 | ••••• | ; | •••••••••••• |
| Д | D | ••••• | 4 | ••••• | ! | •••••••••••• |
| Е | E | ••••• | 5 | ••••• | ? | •••••••••••• |
| Ж | V | ••••• | 6 | ••••• | = | •••••••••• |
| З | Z | ••••• | 7 | ••••• | «» | •••••••••••• |
| И | I | ••••• | 8 | ••••• | ' | •••••••••••••• |
| Й | J | ••••• | 9 | ••••• | () | •••••••••••••• |
| К | K | ••••• | 0 | ••••• | /, № | •••••••••••• |
| Л | L | ••••• | 1 | ••••• | - | •••••••••••• |
| М | M | ••••• | 2 | ••••• | — | •••••••••••• |
| Н | N | ••••• | 3 | ••••• | @ | •••••••••••••• |
| О | O | ••••• | 4 | ••••• | ← | •••••••••••• |
| П | P | ••••• | 5 | ••••• | | |
| Р | R | ••••• | 6 | ••••• | | |
| С | S | ••••• | 7 | ••••• | | |
| Т | T | ••••• | 8 | ••••• | | |
| У | U | ••••• | 9 | ••••• | | |
| Ф | F | ••••• | | | | |
| Х | H | ••••• | | | | |
| Ц | C | ••••• | | | | |
| Ч | | ••••• | | | | |
| Ш | | ••••• | | | | |
| Щ | Q | ••••• | | | | |
| Ы | Y | ••••• | | | | |
| Ь | X | ••••• | | | | |
| Э | | ••••• | | | | |
| Ю | | ••••• | | | | |
| Я | | ••••• | | | | |

ASCII коды (Ақпарат алмасуға арналған америкалық стандартты код) – есептеу техникасында ең көп пайдаланылатын код.

Кодтау кестесінің 16 жолы және 16 бағаны бар. Кестедегі символдың орны оның он алтылық коды арқылы анықталады. Кесте стандартты және баламалы деп аталатын 2 бөліктен тұрады.

ASCII (American Standard Code for Information Interchange) коды стандартты бөлігі

| Код | 16-лық код | Символ | Код | 16-лық код | Символ | Код | 16-лық код | Символ | Код | 16-лық код | Символ |
|-----|------------|--------|-----|------------|--------|-----|------------|--------|-----|------------|--------|
| 0 | 0 | NUL | 32 | 20 | (sp) | 64 | 40 | @ | 96 | 60 | ` |
| 1 | 1 | SOH | 33 | 21 | ! | 65 | 41 | A | 97 | 61 | a |
| 2 | 2 | STX | 34 | 22 | " | 66 | 42 | B | 98 | 62 | b |
| 3 | 3 | ETX | 35 | 23 | # | 67 | 43 | C | 99 | 63 | c |
| 4 | 4 | EOT | 36 | 24 | \$ | 68 | 44 | D | 100 | 64 | d |
| 5 | 5 | ENQ | 37 | 25 | % | 69 | 45 | E | 101 | 65 | e |
| 6 | 6 | ACK | 38 | 26 | & | 70 | 46 | F | 102 | 66 | f |
| 7 | 7 | BEL | 39 | 27 | ' | 71 | 47 | G | 103 | 67 | g |
| 8 | 8 | BS | 40 | 28 | (| 72 | 48 | H | 104 | 68 | h |
| 9 | 9 | TAB | 41 | 29 |) | 73 | 49 | I | 105 | 69 | i |
| 10 | A | LF | 42 | 2A | * | 74 | 4A | J | 106 | 6A | j |
| 11 | B | VT | 43 | 2B | + | 75 | 4B | K | 107 | 6B | k |
| 12 | C | FF | 44 | 2C | , | 76 | 4C | L | 108 | 6C | l |
| 13 | D | CR | 45 | 2D | - | 77 | 4D | M | 109 | 6D | m |
| 14 | E | SO | 46 | 2E | . | 78 | 4E | N | 110 | 6E | n |
| 15 | F | SI | 47 | 2F | / | 79 | 4F | O | 111 | 6F | o |
| 16 | 10 | DLE | 48 | 30 | 0 | 80 | 50 | P | 112 | 70 | p |
| 17 | 11 | DC1 | 49 | 31 | 1 | 81 | 51 | Q | 113 | 71 | q |
| 18 | 12 | DC2 | 50 | 32 | 2 | 82 | 52 | R | 114 | 72 | r |
| 19 | 13 | DC3 | 51 | 33 | 3 | 83 | 53 | S | 115 | 73 | s |
| 20 | 14 | DC4 | 52 | 34 | 4 | 84 | 54 | T | 116 | 74 | t |
| 21 | 15 | NAK | 53 | 35 | 5 | 85 | 55 | U | 117 | 75 | u |
| 22 | 16 | SYN | 54 | 36 | 6 | 86 | 56 | V | 118 | 76 | v |
| 23 | 17 | ETB | 55 | 37 | 7 | 87 | 57 | W | 119 | 77 | w |
| 24 | 18 | CAN | 56 | 38 | 8 | 88 | 58 | X | 120 | 78 | x |
| 25 | 19 | EM | 57 | 39 | 9 | 89 | 59 | Y | 121 | 79 | y |
| 26 | 1A | SUB | 58 | 3A | : | 90 | 5A | Z | 122 | 7A | z |
| 27 | 1B | ESC | 59 | 3B | ; | 91 | 5B | [| 123 | 7B | { |
| 28 | 1C | FS | 60 | 3C | < | 92 | 5C | \ | 124 | 7C | |

| | | | | | | | | | | | |
|----|----|----|----|----|---|----|----|---|-----|----|-----|
| 29 | 1D | GS | 61 | 3D | = | 93 | 5D | J | 125 | 7D | } |
| 30 | 1E | RS | 62 | 3E | > | 94 | 5E | ^ | 126 | 7E | ~ |
| 31 | 1F | US | 63 | 3F | ? | 95 | 5F | | 127 | 7F | DEL |

Стандартты бөліктегі бастапқы 128 символ, яғни 0-ден 127-ге дейінгісі цифрлардың, латын алфавитінің әріптері мен компьютер жұмысын басқаратын арнайы символдардың кодтауы.

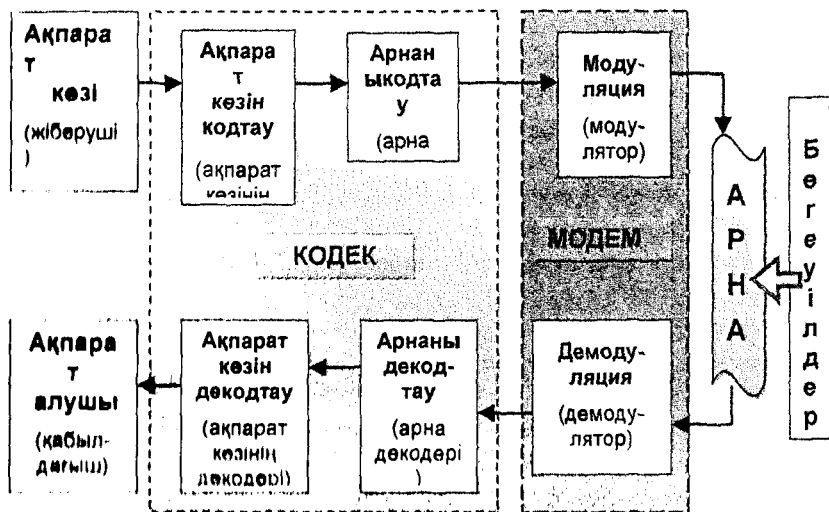
Баламалы бөліктегі қалған 128 символ, яғни 128-ден 255-ке дейінгі кодтар ұлттық әліпби әріптеріне, псевдографикалық және кейбір арнайы символдарды орналастыруға арналған.

Бодо коды — бір-бірінен өзгеше болатын электр сигналдарын кодданатын теңкалыпты телеграфтық 5-биттік код, француз өнертапқышы Эмиль Бодо (J.M.E.Baudot) ұсынған, техника тарихындағы ақпаратты екілік кодтаудың алғашқы тәсілі. Бодо идеясының нәтижесінде әріптерді жіберу және басып шығару процесін автоматтандыру мүмкін болды. Пернелік телеграфтық аппарат жасалды. Бұл ақпаратты қашыққа жіберудегі XIX ғасырдағы техника жетістігі болды. Әріп пернесін басқанда сәйкес 5 импульсті сигнал жасалып, байланыс желісімен жіберіледі. Қабылдаушы аппарат осы сигнал бойынша қағаз лентаға сол әріпті басып отырады.

| Басқару символдары | | | |
|--------------------|---------------------------------|-----------------|---------------|
| 0. ... | Бос орын, әріптер кестесіне өту | | |
| .0 ... | Бос орын, цифрлар кестесіне өту | | |
| 00 ... | Соңғы таңбаны жою | | |
| Әріптер кестесі | | Цифрлар кестесі | |
| .. 0.. | A | 00 0.. | K |
| .. 00. | E | 00 00. | L |
| .. .0. | E | 00 .0. | M |
| .. .00 | I | 00 .00 | N |
| .. 000 | O | 00 000 | P |
| .. 0.0 | U | 00 0.0 | Q |
|0 | Y | 00 ..0 | R |
| .0 ..0 | B | 0. ..0 | S |
| .0 0.0 | C | 0. 0.0 | T |
| .0 000 | D | 0. 000 | V |
| .0 .00 | F | 0. .00 | W |
| .0 .0. | G | 0. .0. | X |
| .0 00. | H | 0. 00. | Z |
| .0 0.. | J | 0. 0.. | — |
| .. 0.. | 1 | 0. 0.. | . |
| .. .0. | 2 | 0. .0. | ^{9/} |
|0 | 3 | 0. ..0 | ^{7/} |
| .. 0.0 | 4 | 0. 0.0 | ^{2/} |
| .. 000 | 5 | 0. 000 | ' |
| .. 00. | ^{1/} | 0. 00. | : |
| .. .00 | ^{3/} | 0. .00 | ? |
| .0 0.. | 6 | 00 0.. | (|
| .0 .0. | 7 | 00 .0. |) |
| .0 ..0 | 8 | 00 ..0 | - |
| .0 0.0 | 9 | 00 0.0 | / |
| .0 000 | 0 | 00 000 | + |
| .0 00. | ^{4/} | 00 00. | = |
| .0 .00 | ^{5/} | 00 .00 | £ |

1.1.3 Цифрлық байланыс арнасында ақпараттың жіберілуі

Төмендегі 1.1-ші суретте цифрлық байланыс арнасында ақпаратты жіберу процесінің моделі көрсетілген.



1.1-сурет. Цифрлық байланыс арнасында ақпаратты жіберу процесінің моделі

Цифрлық байланыс арнасында ақпаратты жіберу процесі моделінің элементтеріне сипаттама берейік.

Ақпарат көзі – жіберілетін хабарды жасайтын физикалық объект, жүйе, немесе құбылыс.

Ақпарат көздерінің (хабардың) 4 түрі бар:

- 1) уақыты бойынша үздіксіз болатын (аналогтік) хабарлар;
- 2) уақыты бойынша дискретті болатын (дискреттелген, үздікті) хабарлар;
- 3) деңгейі бойынша дискретті болатын (квантталған) хабарлар;
- 4) уақыты және деңгейі бойынша дискретті (цифрлық) болатын хабарлар.

Бастапқы ақпараттың (аналогтік ақпарат көздерінің) мысалдарына сөйлеу (речь, speech), музыка, сурет, бейне (видео), қоршаған орта параметрлерінің өлшемдері (температура, қысым, жылдамдық, кернеу және т.с.с.) және тағы басқалары жатады.

Байланыс арнасымен жіберу үшін олар электр сигналдарына түрленеді, жіберілетін хабар бұлардың уақыт ішіндегі өзгерісін бейнелейді. Арнаға жіберілетін хабардың көпшілігі сигнал түрінде емес, сандар тізбегі, мәтіндік және басқа да файлдар түрінде болады.

Цифрлық (дискретті) хабар көздеріне мәтіндер, цифрлық деректер, компьютерлік программалар, түрлі цифрлық өлшеулердің параметрлері, цифрлық фотосуреттер, графикалық және т.б. файлдар жатады. Бұл хабарлардың сипаты әртүрлі, соған сәйкес әртүрлі құрылығдар мен әдістер қолданылады. Арна арқылы жіберуге бірдей құрылығдарды (қазіргі ұялы телефондардағыдай) қолданатындай етіп, хабардың барлық түрін (сөйлеу, музыка, сурет, фото, мәтіндік файлдар және т.с.с.) бірыңғай түрге келтіру үшін деректердің әмбебап бейнеленуі *цифрлық бейнелеу* болып табылады. Кез келген хабар түрін белгілі бір шарттарды сақтай отырып, уақыты және деңгейі бойынша дискретті түрге айналдырып, артынан қажет етілген дәлдікпен тура бастапқы қалпына келтіруге болады. (*Ескерту.* Байланыс арналары туралы [8, 10, 16-18] әдебиеттерді қараңыз).

Байланыс арнасы деп ақпаратты жіберушіден ақпаратты қабылдаушыға хабарды бір жақты жіберуге арналған техникалық құралдардың жүйесі мен сигналдардың таралу ортасын айтады.

Ақпарат кодері ақпарат көзін кодтау үшін қолданылады. Бастапқы ақпарат (сөйлеу, музыка, сурет және т.б.) көбінесе адамның сезім мүшелері арқылы қабылдауға арналған, олар байланыс арнасымен тікелей жіберуге келмейді, сондықтан оларды кодтайды.

Ақпарат көзін кодтаудың мақсаты – ақпаратты жіберу процесін оңайлату үшін, сондай-ақ ақпаратты жіберу жылдамдығын арттыру мақсатында ақпараттың көлемін қысқарту немесе жіберуге қажетті жиілік жолағын қысқарту үшін түрлі хабарларды бірыңғай формаға түрлендіру. Ақпарат көзін кодтауды *үнемді кодтау* немесе *тиімді кодтау*, сондай-ақ *деректерді сығу* (немесе *қысқарту*, *ықшамдау*) деп те атайды (3-бөлімде қарастырылады).

Арнадағы кодтау (немесе *бөгөуілге орнықты кодтау*) деп шулы арналарға ақпаратты жібергенде хабардағы қателерді анықтап,

түзете алатындай етіп жіберілетін хабарды түрлендіруді айтады (4-бөлімде қарастырылады).

Ақпарат көздерін кодтау мен арнаны кодтаудың арасында белгілі айырмашылық бар. Ақпарат көздерін кодтаудың мысалына мәтінді Морзе кодымен беру, компакт-дискіге жазу кезінде дыбыстық сигналды цифрлау жатады.

Ақпарат көздерін кодтау кезінде хабардың артықтығы азаяды және мұндай кодтауды әдетте *деректерді сызу* деп атайды.

Ал, *арналарды кодтау*, керісінше, ақпараттың артықтығын ұншайтады, өйткені бастапқы ақпараттық символдарға қосымша *тексеруші* (немесе *бақылаулық*) деп аталатын символдар тіркеледі. Ақпаратқа қосымша тексеруші символдарды енгізу арқылы оны шулы арнамен жіберу кезінде хабарда туындайтын қателерді шыңдауға және түзетуге болады.

Арнаны осылай тексеруші символдар енгізу арқылы кодтауды *бөгеуілге орнықты кодтау* деп атайды. Бөгеуілге орнықты кодтаусыз CD-ROM, DVD немесе қатты дисктер сияқты құрылғыларды жасау мүмкін болмас еді. Бұдан қазіргі заманғы цифрлық техникалар, цифрлық байланыс жүйелері кодтаусыз болмайтындығын байқауға болады.

Модуляция дегеніміз жіберілетін хабар символдарын арнаның талаптарына сәйкес сигналдарға түрлендіру процесі. *Модулятордың* міндеті – кодталған хабарды байланыс арнасында жіберілетін сигналдарға айналдыру.

Демодуляция (сигналды детекторлау немесе сигналдарды қабылдау) – қабылдағыштың арнадағы қабылданған тербелістерінен (*қабылданған тізбек* деп аталатын) кодтық тізбекті алуы.

Қабылдағыштың міндеті пайдалы сигнал мен бөгеуілдің қоспасынан тұратын қабылданған хабардан мүмкіндігінше шынайы бастапқы хабарды қалпына келтіріп алу.

Арна декодері. Қабылданған тізбек арнаға жіберілген тізбекпен бірдей болмауы мүмкін, яғни қатемен қабылдануы мүмкін. *Арна декодерінің* міндеті осы қателерді анықтап, мүмкіндегенше түзету болып табылады. Қабылданған кодтық тізбектегі қатені анықтау және түзету процесі *арнаны декодтау* деп аталады.

Ақпарат көзінің декодері. Арнаға жіберілген ақпарат көзінің хабары жіберу барысында ықшамдалған (сығылған) түрде

кодталғандықтан, оны бастапқы түріне келтіру қажет, осыны *ақпарат көзін декодтау* дейді.

1.1.4 Кодтың негізгі сипаттамалары

Кодтың негізгі сипаттамаларына тоқталайық.

Код сипаттамалары екі топқа бөлінеді:

- 1) құрылымдық сипаттамалар;
- 2) код сигналдарының сипаттамалары.

Құрылымдық сипаттамалар өз ішінен үшке бөлінеді:

- 1) импульстік белгілер саны бойынша;
- 2) кодтық комбинациядағы разрядтар саны бойынша;
- 3) комбинациялау тәсілі бойынша.

Импульстік белгілер саны бойынша бірлік, екілік, көппозициялы кодтар болып бөлінеді.

Кодтық комбинациядағы разрядтар саны бойынша разрядтар саны тұрақты (теңқалыпты, біркелкі) және разрядтар саны тұрақты емес (теңқалыпты емес, әркелкі) кодтар болып бөлінеді.

Комбинациялау тәсілі бойынша кодтарды комбинацияларды толығымен қолданатын және комбинацияларды толық қолданбайтын кодтар деп екіге бөледі.

Кодтың сигналдық сипаттамаларына *импульстік белгілер және жіберу тәсілі* жатады.

Импульстік белгілер бойынша код сигналдары алты түрге бөлінеді: амплитудалық, жиіліктік, полярлық, фазалық, уақыттық, комбинацияланған сигналдар.

Код сигналдарын жіберу тәсілі үшке бөлінеді: тізбекті, параллель, аралас тәсіл.

1.1.5 Кодты сипаттайтын негізгі ұғымдар

Кодтарды олардың әртүрлі сандық және сапалық көрсеткіштерін көрсететін негізгі сипаттамалары арқылы бағалайды. Ақпаратты сақтауға, арнамен жіберуге, өндеуге арналған кодтарды таңдау кезінде осы сипаттамалар қолданылады. Мұндай сипаттамаларға кодтың ұзындығы, код негізі, кодтың қуаты, кодтық комбинациялардың жалпы саны, ақпараттық символдар саны, тексеруші символдар саны, кодтың артықтығы,

кодтық комбинацияның жіберілу жылдамдығы, кодтық комбинацияның салмағы, кодтық қашықтық, кодтың салмақтық сипаттамасы, анықталмаған қатенің ықтималдығы, кодтың оңтайлылығы, жалған өтулердің коэффициенті жатады.

Код ұзындығы (n) – кодтық комбинацияны құрайтын символдар (разрядтар) саны.

Кодтың негізі (m) – кодтық комбинацияларда қолданылатын импульстік белгілердің бір-бірінен өзгеше мәндерінің саны. Мысалы: екілік кодтарда $m=2$, себебі бұл кодта импульстік белгілердің мәні тек екеу: 0 мен 1.

Кодтың қуаты N_k – хабарды жіберу үшін қолданылатын кодтық комбинациялардың (жұмыстық кодтық комбинациялар) саны.

Кодтық комбинациялардың жалпы саны N – барлық мүмкін болатын комбинациялардың саны, $N = m^n$. Екілік кодтар үшін $N=2^n$ болады.

Ақпараттық символдар саны k – хабарды жіберуге арналған кодтық комбинациядағы арнайы символдар (разрядтар) саны, осыдан $N_k=2^k$ болады.

Тексеруші символдар саны r – кодтық комбинациядағы қателерді түзетуге арналған арнайы символдар (разрядтар) саны. Бұл сән кодтың абсолютті артықтығын сипаттайды.

Кодтық артықтығы R деп кодтың салыстырмалы артықтығын айтады, $R=r/n$ (тексеруші символдар санының кодтың ұзындығының қатынасына тең). Толық сипаттамада бұл формула $R=1-\log_m N_k / \log_m N$ түрінде болады. Мысалы: Код ұзындығы $n=5$ болатын 16 хабарды жіберуге арналған кодтың артықтығы мен жылдамдығын есептеу керек. Барлық кодтық комбинация саны $N=2^5=2^5=32$. Кодтың артықтығы $R=1-\log_2 16 / \log_2 32=1-4/5=0.2$. $R'=1-R=1-0.2=0.8$

Кодтық комбинациялардың жіберілу жылдамдығы – ақпараттық символдардың код ұзындығына қатынасына тең: $R'=k/n$.

$n=k+r$ болғандықтан, $R'=1-R$ болады.

Кодтың салмағы ω – кодтық комбинациядағы бірлердің саны.

Мысалы: 1010101, мұнда кодтық комбинация ұзындығы $n=7$, ал комбинациядағы бір саны төртеу, $\omega=4$.

Кодтық қашықтық d – бұл екі кодтық комбинацияның бірдей позицияларындағы бірдей емес символдардың саны. Минималды кодтық қашықтықты d_0 деп белгілейді.

Мысалы: 10010000 және 00000010 кодтық комбинацияларының арасындағы кодтық қашықтықты табу үшін оларды екілік модуль арифметикасы бойынша қосамыз :

$$\begin{array}{r} 10010000 \\ \oplus \quad 00000010 \\ \hline 10010010 \end{array}$$

Екілік модуль бойынша қосу ережесі: $1\oplus 1=0$, $0\oplus 0=0$, $1\oplus 0=1$, $0\oplus 1=1$

Қосу нәтижесінде алынған жаңа кодтық комбинацияның салмағы $\omega=3$, яғни 2 комбинация арасындағы кодтық қашықтық $d=3$.

Кодтың салмақтық сипаттамасы $W(\omega)$ – кодтың салмақтық комбинацияларының саны.

Мысалы: ұзындығы $n=7$ және 1000000, 1010100, 11100000 кодтық комбинациялардан тұратын код үшін салмақтық сипаттамалар:

$W(1) = 1$ (салмағы $\omega=1$ комбинация біреу ғана) 1000000 ($\omega=1$);

$W(2) = 0$ (салмағы $\omega=2$ комбинация жоқ);

$W(3) = 2$, (салмағы $\omega=2$ екеу) 1010100 ($\omega=3$); 11100000 ($\omega=3$);

Анықталмаған қатенің ықтималдығы $P_{ақ}$ – бұл қабылданған кодтық комбинация жіберілген кодтық комбинациядан өзгеше болатын және де берілген кодтың қасиеті қатенің бар екендігін анықтауға мүмкіндік бермейтін оқиғаның болу ықтималдығын білдіреді.

Кодтың оңтайлылығы (оптимальность) – ұзындығы n -ге, артықтығы g -ге тең барлық кодтардың ішінде анықталмаған қате ықтималдығының ең аз мәнін қамтамасыз ететін кодтың қасиеті.

Жалған өтулердің коэффициенті мына формуламен (1) анықталады :

$$K_{жалған}(d) = \frac{1}{N_{жс}} \sum_i^N \frac{N_{жс}^i(d)}{C_n^d} \quad (1)$$

мұндағы $N_{ж}^i(d)$ – i -ші кодтық комбинациядан d кашықтықта
 гүрған жұмысшы кодтық комбинациялардың саны; ал C_n^d – бұл n
 комбинациядан d түрлі нұсқалар. Бұл коэффициент d еселікте
 китенің қандай бөлігі анықталмайтындығын көрсетеді.

1.1.6 Кодтарды бейнелеу тәсілдері

Кодтарды бейнелеудің қарапайым тәсілі – оны *кесте түрінде*
 жизу (1.1-ші кесте). Кодтық кесте арқылы кодталатын бір хабарға
 (λ_i) әртүрлі кодтарды сәйкес қоюға болады, олардың жасалу
 жолдары әртүрлі.

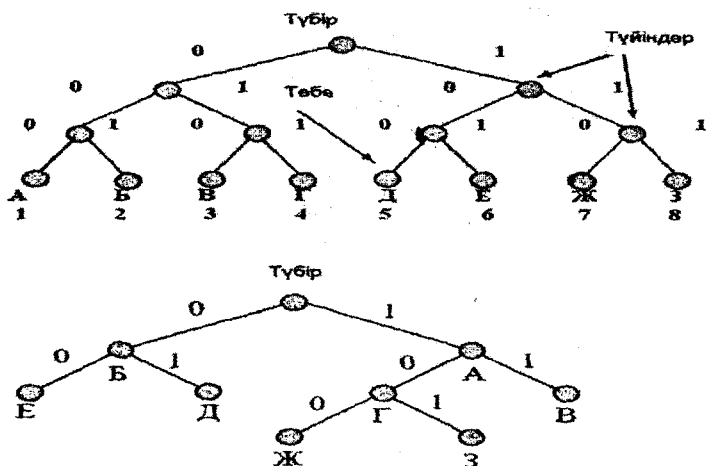
Кесте 1.1– Кодты кесте түрінде бейнелеу

| Әріп λ_i | Сан λ_i | Негізі 4 болатын код (біркелкі код) | Негізі 2 болатын Код1 (әркелкі код) | Негізі 2 болатын Код2 (префикстік код) |
|---------------------|--------------------|---|---|--|
| А | 0 | 00 | 000 | 0 |
| Б | 1 | 01 | 001 | 10 |
| В | 2 | 02 | 010 | 1100 |
| Г | 3 | 03 | 011 | 1101 |
| Д | 4 | 10 | 100 | 1110 |
| Е | 5 | 11 | 101 | 11110 |
| Ж | 6 | 12 | 110 | 111110 |
| З | 7 | 13 | 111 | 111111 |

Кодтарды бейнелеудің (берудің) келесі бір көрнекі, әрі тиімді
 тәсілі – бұл оны λ_i кодтық символға сәйкес келетін «түбір»,
 «бұтақтар», «түйіндерден» тұратын «кодтық ағаш» түрінде
 бейнелеу (1.2-сурет).

Кодтық ағаш төбелерінде ақпарат көзінің әліпбиінің әріптері
 орналасады, әрбір әріптің өз төбесі және түбірден төбеге дейін өз
 жолы болады. Кодтық ағашты қолданып кодтау түбірден төбеге
 қарай қандай-да бір жолмен жүруден тұрады, мұнда оңға қарай
 жүруге «1», солға жүруге «0» сәйкес келеді. Мысалы, 1.2.-суреттегі
 жоғарғы «кодтық ағашта» А әрпіне 000 коды, Д әрпіне 110 коды
 сәйкес келеді. Бұл «кодтық ағашта» үш разрядты теңқалыпты
 (біркелкі, равномерный) код бейнеленген. Ал төменгі «кодтық

ағаш» теңқалыпты емес (эркелкі, неравномерный) кодты бейнелеген, өйткені кодтық комбинациялардағы разрядтар саны әртүрлі, А әрпіне (төбесіне) 1 коды, Д әрпіне 01 коды, Ж әрпіне 100 коды сәйкес келеді.



1.2-сурет. Кодты «кодтық ағаш» түрінде бейнелеу

1.2 Кодтау теориясының даму тарихы

Кодтау теориясының тарихына үнілсек, санау жүйелері алғашқы кодтар болып саналады. Кодтау теориясының келесі бір бағыты – бұл криптография немесе құпия кодтау. Қазіргі заманғы кодтау теориясының дамуы байланыс жүйесінің дамуымен тікелей байланысты болды. Энтропияға негізделген Шеннонның ақпарат теориясы ақпаратты сығуда қолданылатын тиімді кодтардың (Шеннон-Фано, Хаффман кодтары) математикалық негізі болып табылады. Арнадағы шулардан ақпаратты және байланыс жүйесін қорғау қажеттігі артықтығы бар кодтар (Хемминг коды, циклдік кодтар) теориясының дамуына әкелді. Осылайша, қазіргі заманғы кодтау теориясы төрт түрлі даму бағытын қамтиды: санау жүйелерінің теориясы; криптография теориясы; тиімді кодтар теориясы; артықтығы бар кодтар теориясы. Кодтауда

математикалық аппарат ретінде сандар теориясы, топтар теориясы аппараты, матрицалар теориясының аппараты және т.б. қолданылады.

Ақпаратты артықтықпен (бөгеуілге орнықты) кодтаудың теориясы мен практикасының пайда болуы мен дамуын 1946 жылы АҚШ ғалымы Клод Шеннонның «Ақпарат теориясы мен кибернетика бойынша жұмыстар» монографиясының жарияланғанымен байланыстырады [8]. Бөгеуілге орнықты кодтауда қатені түзететін кодтарды қолдану арқылы ақпарат жіберудің шынайылығы қамтамасыз етіледі. Бұл жұмыста Шеннон ақпарат жіберудің жылдамдығы байланыс арнасының өткізу қабілетінен аз болса, онда ақпаратты қате қабылдау ықтималдығы өте аз болуын қамтамасыз ететіндей артықтығы бар (бөгеуілге орнықты) кодты табуға және осы кодпен ақпаратты кодтау тәсілін (кодтау алгоритмі қарастырылмаған) табуға болатынын дәлелдеген.

Шеннон бұл жұмыста аса жақсы сипаттамалары бар арнаны жасау қымбатқа түсетіндіктен, экономикалық тұрғыдан артықтығы бар (бөгеуілге орнықты) кодтарды қолдану тиімді екенін көрсеткен. Алайда, мұндай кодтарды қалай құруды көрсетпеген, олардың тек бар екенін айтқан. Шеннон жұмысының нәтижелері Хинчин, Колмогоров, Харкевич, Варшамов сияқты және т.б. кеңестік ғалымдардың жұмыстары негізінде жасалған [8].

Көпшілік әдебиеттерде кодтау теориясының тарихы 1948 жылы Клод Шеннонның әйгілі «Хабарларды жіберудің математикалық теориясы» атты мақаласы жарияланғаннан бастау алады деп айтылады. Мақалада егер арнаның өткізу қабілеті хабар көзінің энтропиясынан артық болса, онда хабарды артықтықпен тоқтаусыз кодтауға болатындығы көрсетілген. Басқаша айтсақ, егер хабардың жіберілу жылдамдығы байланыс арнасының өткізу қабілетінен аз болса (көп болған жағдайда қабылдаушы құрылғының кірісінде сигнал/шу қатынасы жоғары болады), онда сигналдарды қатесіз жіберуге болатындығы дәлелденген. Бұл байланыс желісінің энергетикасы олардың тек өткізу қабілеттілігін ғана анықтайтындығын, ал хабарды қабылдау кезінде арнайы үлгі бойынша құрылған кодтарды қолдану арқылы жоғары деңгейдегі бөгеуілге орнықтылыққа жетуге болатындығын көрсетеді. Шеннонның бұл ғылыми идеясы байланыс теориясына, сигналдарды жіберу теориясына да әсерін тигізді. Өйткені, бұл

теорияға дейін сигналдарды қабылдау кезінде бөгеуілге орнықтылықты жоғарылатудың жолдары ретінде (1-әдіс) жіберушінің қуатын арттыру әдісі немесе (2-әдіс) сол хабарды байланыс арнасымен бірнеше рет жіберу әдісі қолданылған болатын. Аталған екі әдіс те байланыс арнасының өткізу қабілеттілігін қолданудың төмен тиімділігіне алып келеді.

XX-ғасырдың 50-ші жылдарында кодтау теориясының келесі мәселелерін шешуге бағытталған көптеген зерттеулер жүргізілді:

- жақсы түзетуші қасиеттері бар кодтарды құру;
- декодтаудың қарапайым алгоритмін таңдау;
- қателерді түзетуші кодтардың, модуляция түрлерінің, декодтау алгоритмдерінің және байланыс арнасының сипаттамаларының сәйкестігі.

Шеннонның жұмысы жарияланғаннан кейін ғалымдар бөгеуілге орнықты кодтарды, ақпаратты кодтау тәсілдерін іздеумен шұғылданды. 1948 жылдың соңында бөгеуілге орнықты кодтау теориясы мен практикасы 2 бағытта дамыды [6], [21], [22]:

1-ші бағыт алгебралық сипатта болды, негізінен блоктық кодтарды құру тәсілдері мен оларды декодтау алгоритмдері қарастырылды.

2-ші бағыттағы зерттеулерде бөгеуілге орнықты кодтау мен декодтау ықтималдық тұрғысынан қарастырылды. Алғашқы зерттеулер үздік блоктық кодтардағы қатенің ықтималдықтарын бағалаумен байланысты болды. Бұл зерттеулер үздіксіз кодтарды дамытуға, жималаушы кодтарға жататын кодтардың ашылуына себепкер болды.

Кодтау теориясындағы кодтау және декодтау әдістерінің (алгоритмдерінің) жылдар бойынша шығу тарихына қысқаша шолу жасайық.

1949 жылы Голейдің (АҚШ) сызықтық блоктық кодтары құрылды.

1949 жылы Хэмминг екі блоктық кодты $(n_1, k_1, d_{o1}) = (7, 4, 3)$ және $(n_2, k_2, d_{o2}) = (7, 3, 4)$ ашты, сызықтық блоктық кодтар теориясын жасады, блоктық кодтардың негізгі параметрлеріне анықтама берді, өз кодтары үшін кодтаушы (кодер) және декодтаушы (декодер) құрылғылар жасады. Кодтың түзету қабілетін бағалауға арналған кодтық қашықтық (d) және минимальды кодтық қашықтық (d_o) параметрлерін енгізді

1950 жылы Хэммингтің (АҚШ). бір қатені түзететін сызықтық кодтары құрылды.

М.Дж.Голей мен Р.Хэмминг құрған бір қатені түзететін алғашқы блоктық кодтарда ақпараттық символдар тізбегі берілген ұзындықтағы жеке блоктарға бөлінеді. Мұндай әрбір блокқа қосымша тексеруші символдар тіркеледі, олар берілген ақпараттық символдарды екілік модуль бойынша қосу арқылы жасалады. Ақпараттық және тексеруші символдардан тұратын блок байланыс арнасына жіберіледі. Хэмминг кодының декодтау алгоритмі қарапайым, қабылданған кодтық комбинациядағы бір қатені түзетуге мүмкіндік берді. Бұл кодтар Шеннонның теоремасымен салыстырғанда әлдеқайда әлсіз болғанымен, бірақ олардың маңызы өте жоғары. Алғаш рет (1950 жылы) Хэмминг ұсынған қатені түзететін кодтардың (Error-Correcting Code, ECC) қазіргі кездегі модификациялары деректерді сақтаудың барлық жүйелерінде және процессор мен жедел жад арасында ақпарат алмасуда қолданылады.

1954 жылы Р.А.Вагнер кодына арналған бөгеуілге орнықтылықтың оңай шешіміне зерттеу жасалды (АҚШ ғалымдары Сильверман және М.Болсер).

Кез-келген цифрлық жүйенің негізі аналогтық арна болып табылады. Мұндай арнада цифрлық байланыс жүйесі модулятор/демодулятордан (модем) тұрады, ол аналогтік арнаны кодерді/декодерді (кодек) қолдана алатындай дискреттік арнаға түрлендіреді (1.1.3-бөлімде айтылған).

Қатенің ең аз ықтималдығын, қабылданған сигнал және барлық мүмкін болатын кодтық комбинациялар арасындағы қашықтықты демодуляторда есептеу арқылы алуға болады. Қабылданған сигналдан аз қашықтықта орналасқан кодтық комбинацияның пайдасына шешім қабылданады. Бірақ аса көп кодтық комбинациялардан тұратын ұзын кодтар үшін демодулятордың күрделілігі жоғарылайды, мұндай жағдайда оны іс жүзінде іске асыру мүмкін емес. Сондықтан кодтарды қолдану модем мен кодектің бір-бірімен келіскен жағдайында жақсы нәтижелерді береді. Егер қабылданған символға демодуляторда жеңіл шешім қабылданса және егер декодер демодуляторда қабылданған шешімнің сенімділігін сипаттайтын қосымша ақпаратты алса, онда жоғарыда айтылған тұжырым мүмкін болады. Қарапайым жағдайда,

екілік сигналдарды жіберу кезінде демодуляторда қабылданған, бірақ сенімді шешім қабылданбайтын символдар өшіріледі және декодерге жойылған символдары бар кодтың позициялары туралы хабар беріледі.

Кодтық комбинацияларды декодтау үшін демодуляторда жеңіл шешім қабылдау идеясы кодтау теориясының бастапқы кезеңдерінде туындаған болатын. Америкалық ғалымдар Сильверман мен Болсердің 1954 жылы жасаған зерттеулері бұл бағыттағы алғашқы жұмыстарға жатады. Олар бөгеуілге орнықтылықты Вагнер кодын жеңіл шешу арқылы зерттеді, бұл код ақпараттық символдарды жұптылыққа тексеру үшін бір ғана артықтығы бар екілік символдан тұрды. Қабылдағышта барлық қабылданған символдардың мәндері туралы қатаң шешім қабылданады және бұған қоса қосымша олардың арасынан ең сенімдісі анықталады. Декодтау барысында бұл символ, егер қабылданған кодтық комбинация жұптылыққа тексерілмеген жағдайда кері символға өзгереді. Мұндай декодтау алгоритмі кодтық комбинацияда жалғыз тексеруші символ болған кезде бір рет кездесетін қателерді түзетуге мүмкіндік береді. Бұл идея кейін сенімсіз қабылданған символдарды өшіру арнасында, БЧХ кодын декодтауда қолданылды.

1956 жылы Д.А.Хаффман (АҚШ) сызықтық желілерді сызықтық фильтр теориясы тұрғысынан зерттеп және қателерді түзететін кодтарды алу үшін жылжыту регистрі бар генераторларды қолдануды ұсынды.

1956 жылы америкалық ғалым Д. Слейян топтар теориясының математикалық аппаратын қолданып алғаш рет жұптылыққа тексеруі бар сызықтық блоктық кодтардың негізін қалады.

1960 жылы тиімді сызықтық блоктық кодтар құрылды. Оны АҚШ ғалымдары И.С.Рид және Г.Соломон ұсынды. ЕСС бір нұсқасы болып табылатын Рид-Соломон коды компакт-дискілерде қолданылады.

Хэмминг кодына ұқсас кодтардың версиялары көп, олардың айырмашылықтары кодтау алгоритмдері мен тексеруші биттердің (разряд) санында ғана. Планетааралық станциялармен ғарыштық байланыс жасауда қолданылатын Рид-Мюллер кодында жеті ақпараттық разрядтарға (биттерге) 32 тексеруші, ал алты ақпараттық разрядтарға 26 тексеруші разряд тіркеледі.

Бұл кодтардың құрылуы алгебралық кодтау теориясы саласында көптеген зерттеулерге жол ашты. Хэммингтің ғылыми еңбегі осы салада 1954 жылы басталған көптеген жаңа идеяларға түрткі болды.

1954 жылы мажоритарлы декодтауы бар кодтар құрылды (АҚШ ғалымдары И.С.Рид және Д.Е.Маллер ұсынды). Рид-Маллер кодының мажоритарлы декодтауын алғаш америкалық ғалым И.С.Рид ұсынды. Мажоритарлы декодтау барысында әрбір ақпараттық символ үшін қабылданған кодтың анықталған символдар комбинациясын екілік модуль бойынша қосу арқылы плынған бағалардың тақ саны қалыптасады. Қабылданған символдың ақиқаттығы туралы шешім мажоритарлы принцип бойынша қабылданады. Егер бағалар санының көп бөлігі 1-ге тең болса, онда тура осындай шешім қабылданады.

1963 жылы демодулятордың жеңіл және қатаң шешімдерінде блоктық кодтарды декодтаудың мажоритарлы әдісі зерттелді. АҚШ ғалымы Дж.Л.Месси осыған ұқсас кодтарды құру мен декодтаудың жалпы принциптерін бекітті.

1965 жылы мажоритарлы декодтау бойынша циклдік кодтар зерттелді. КСРО ғалымдары В.Д.Колесников пен Е.Т.Мирончиков мажоритарлы декодтауы бар кодтардың құрылуына айтарлықтай үлес қосты. Мажоритарлы декодтауы бар кодтардың декодтау алгоритмдері қарапайым және жылдам болғанымен, мұндай кодтар өте аз және басқа кодтарға қарағанда әлсіз болып табылады.

1955 жылы алғаш рет жималаушы кодтар құрылды (АҚШ ғалымы П.Элайес, КСРО ғалымдары Л.М.Финк және В.И.Шляпоберский ұсынды).

1955 жылы АҚШ және КСРО-да қазіргі заманғы байланыс техникасында кеңінен қолданылатын жималаушы және рекуррентті кодтардың маңызды класы ұсынылды. Осындай кодтарды құру және оларды декодтау алгоритмін есептеуді тиімділеу бойынша жиырма жыл зерттеулер жасалды.

Жималаушы кодтар блоктық сызықтық кодтардың жеке жағдайы болып табылады. Блоктық сызықтық кодтарға жималаушы құрылымды енгізу арқылы оның декодталуын жеңілдетуге болады. Мұндай кодтардың құрылымы ағаш немесе тор (желі) түрінде болады. Ағаш тәрізді құрылымның әрбір қабырғасына m

ақпараттық символдардың анықталған тізбегі сәйкес келеді. Қабылданған символдар тізбегі бойынша әрбір қабырғаға оның салмағы – қабылданған тізбектен оның қашықтығын сипаттайтын сан табылуы мүмкін. Бұл қашықтықты өлшеу үшін келесі 2 метрика қолданылады: 1) егер демодуляторда қатаң шешім қабылданса Хэмминг метрикасы немесе 2) барынша шындыққа ұқсастық әдісі бойынша декодтау жүргізілсе Евклид метрикасы қолданылады. Жималаушы кодтарды декодтау қабылданған символдар тізбегінен қашықтығы аз болатын жолдарды кодтық тор бойынша жүйелеуден тұрады. Кодтың жималаушы құрылымы осы қашықтықты есептеуді жеңілдететін, рекурентті алгоритмдерді қолдануға мүмкіндік береді.

1957 жылы жималаушы кодтарды декодтау үшін тізбекті декодтау әдісін АҚШ ғалымы Дж.Возенкрафт ұсынды. Бұл жағдайда декодерде жималаушы кодтың кодтық торының қабырғасында барлық мүмкін жолдар емес, тек неғұрлым ықтималды жолдар ғана қарастырылады. Егер декодер қандай да бір қадамда қате жолды таңдаса, онда ол келесі қабырғаларды таңдау барысында таңдалған жол мен қабылданған тізбек арасындағы қашықтықтың тез артатынын анықтайды. Бұл декодерге бірнеше рет артқа қадам жасап және максималды түрде ұқсас альтернативті жолдарды зерттеуге арналған белгі болып табылады. Тізбекті декодтау барысында бір қабырғаға есептелген сан кездейсоқ шама болады және декодер жадысында барлық зерттелген бұтақтар үшін есептелген қашықтық сақталуы қажет.

1961 жылы тізбекті декодтау әдісінің тиімділігі зерттелді. Тізбекті декодтау алгоритміне алғашқы зерттеулерді АҚШ ғалымдары Дж. Возенкрафт және Б. Рейффен жүргізді.

1963 жылы тізбекті декодтау алгоритмінің модификациясы құрылды. Тізбекті декодтау алгоритмін АҚШ ғалымы Р.М.Фано жетілдірді.

1966 жылы тізбекті декодтаудың стэк-алгоритмі құрылды. 1966 жылы осы тізбекті декодтау алгоритмінің тиімді түрін кеңестік ғалым К.Ш.Зигангиров ұсынса, ал бұдан кейін 1969 жылы американдық ғалым Ф.Джелинек осыған ұқсас алгоритмді ұсынды.

1967 жылы жималаушы кодтарды максималды ұқсастық әдісі бойынша декодтау алгоритмі құрылды. АҚШ ғалымы А.Дж.Витерби ұсынған жималаушы кодтарды декодтау алгоритмі (Витерби алгоритмі) кодтау теориясы саласындағы маңызды

жетістіктердің бірі болды. Бұл алгоритмнің тізбекті декодтау алгоритмінен айырмашылығы мынада: k кодтық шектеу ұшындығында кодтық тор бойынша барлық мүмкін жолдар зерттеледі, сондықтан ол шағын мәнді жималаушы кодтарды декодтауға қолайлы.

Жималаушы кодтар, Витерби алгоритмі және тізбекті декодтау әдістері қазіргі кезде магистральды радиорелелік және жерсеріктік байланыс жүйелерінде кең тараған.

1957 жылы циклдік кодтар алғаш рет зерттелді. Америкалық ғалым Е.Прейндж алғаш рет циклдік кодтар ұғымын енгізді. Циклдік кодтар – сызықтық кодтардың маңызды ішкі класы. Циклдік кодтардың кодтау мен декодтау алгоритмдері Галуа өрісінің алгебралық теориясын қолдануға негізделгендіктен тиімді болып табылады. Циклдік кодтар теориясын америкалық ғалымдар Питерсон, Берлекамп және Касами дамытты.

1959 жылы БЧХ кодтары құрылды. 1960 жылы АҚШ ғалымдары Р.К.Боуз және Д.К.Рой-Чоудхури және 1959 жылы Франция ғалымы А.Хоквингем ұсынған БЧХ коды – сызықтық циклдік кодтардың маңызды және кең тараған түрі, қабылданған кодтық комбинациядағы көптеген қателерді түзете алады. БЧХ деп кодты ашқан Боуз, Рой-Чоудхури, Хоквингем аттарының алғашқы әріптеріне байланысты аталды.

1960 жылы ақпараттық символдарды қайта ауыстыру әдісін қолдану арқылы қателер топтары бар арнада сигналдардың тәуелсізге орнықтылығына зерттеу жасалды (1960 жылы КСРО ғалымдары А.А. Харкевич және Э.Л.Блох, 1970 жылы АҚШ ғалымдары Дж.Л.Рамсей).

Қателер дәстесі (пакет) туындайтын байланыс арнасында жұмыс істеуге арналған арнайы кодтар құрылды. Жіберілген кодтық комбинациядағы көршілес символдарды байланыс арнасында олардың бүлінуі тәуелсіз болатындай етіп, уақыт ішінде жеткілікті деңгейде үлестіру қолданылады. Қабылдағышта символдардың бастапқы реті қалпына келтіріледі. Бұл қарапайым әдісті 1960 жылы кеңестік ғалымдар А.А.Харкевич және Э.Л.Блох, кейінірек 1970 жылы америкалық ғалым Дж. Л. Рамсей ашты. Бұл қазіргі кезде цифрлік сигналдарды қабылдау кезінде қателер тобы пайда болатын радиорелелік және жылжымалы байланыс жүйелерінде қолданылады.

1964 жылы каскадтық кодтар құрылып, зерттеулер жасалды (Г.Форни, Э.Л.Блох, В.В.Зяблов). АҚШ ғалымы Г.Форни блоктық кодтардың бір класын – каскадтық кодтарды ұсынды, зерттеу барысында Г.Форни және кеңестік ғалымдар Э.Л.Блох, В.В.Зяблов аса маңызды нәтижелер алды. Э.Л.Блох пен В.В.Зяблов зерттеулері (1976 жыл, 1982 жыл) жоғарыда айтылған бөгеуілге орнықты кодтау мәселелерін шешу үшін каскадтық кодтар ішкі және сыртқы кодтарды сәйкесінше дұрыс таңдау арқылы мүмкіндік беретіндігін көрсетті.

1960 жылдардың соңы мен 1970 жылдың басында сигналдарды тұтас қабылдау әдістерін өңдеуге бағытталған көптеген зерттеулер жасалды. Қабылдағыш құрылымында кодтық комбинацияны қабылдау жөніндегі шешім максималды ұқсастық әдісі (метод максимального правдоподобия) бойынша қабылданады, яғни декодтау кезінде қабылданған сигналдан ең аз евклидтік қашықтықта орналасқан кодтық комбинация («ең ұқсасы») ізделінеді. Кодтың құрылымымен ескеру осы қашықтықты оңай есептеуге мүмкіндік береді.

1971-1972 жылдары блоктық кодтарды жеңіл декодтау алгоритмі құрылды. АҚШ ғалымдары Е.Велдон мен Д.Чейз блоктық кодтар үшін демодулятордың жеңіл шешімін қолданатын декодтау алгоритмін ұсынды.

1980 жылдан бастап сигналды-кодтық конструкция деп аталатын арнайы құрылған сигналдарды өңдеуге және талдауға арналған жаңа ғылыми бағыттар пайда болды. Бұл бағыттар модуляция және кодтау әдістерінің синтезін және олардың оңтайлы жұмыстарын ұсынды. Сонымен бірге, шектеулі жиіліктік жолағы бар байланыс арнасы арқылы ақпаратты сапалы түрде жіберуге болатын, Котельников және Шеннон теорияларының қағидаларымен анықталатын байланыс жүйесін құруға мүмкіндік береді.

1982 жылы торлы кодтық модуляция зерттелді. АҚШ ғалымы Г.Унгербок теориялық зерттеулер нәтижесінде торлы кодтық модуляция (ТКМ) құрды. Модуляцияның бұл түрі көп позициялы сигналдар мен бөгеуілге орнықты кодтардың үйлесімдігіне негізделеді. Символдардың ақпараттық тізбегі жималаушы кодтардың көмегімен кодтық тізбекке түрленеді. Кодталған биттер ішкі ансамбльдің таңдалуын, ал кодталмағандары осы таңдалған ішкі ансамбльдегі нақты сигналдық нүктені анықтайды.

Сигналдарды оңтайлы қабылдау үшін торлы кодтық модуляцияда Питерби алгоритмі арқылы жүзеге асатын, барынша шындыққа үкенту әдісі қолданылады. Торлы кодтық модуляцияны қолдану арқылы қажетті желі энергиясын және жиіліктік жолақтарды ашытуға болады. Бұл модуляция түрі магистральды жоғары жылдамдықтағы радиорелелік желілер мен жерсеріктік байланыстарда қолданылады.

1982-1991 жылдары гаусстық байланыс арнасындағы хабарды қабылдау кезіндегі *әр түрлі сигналды-кодтық конструкция әдістерінің бөгеуілге орнықтылығын* КСРО ғалымдары С.М.Портной, В.В.Зяблов, В. В.Гинзбург, В.Л.Банкет зерттеді.

1993 жылы *турбокодтар құрылды (К.Берроу және А.Главью ұсынды)*. К.Берроу бастаған француз ғалымдары турбокодтар деп аталатын параллельді каскадтық кодтың жаңа класын ұсынды. Турбокодтарды үлкен ұзындықтағы блоктық кодтар ретінде ирестіруға болады. Хэмминг, БЧХ, Рид-Соломон кодтары турбокодтың компоненттері болып саналады. Соңғы жылдары бұл кодтар қарқынды түрде дамып келеді және кеңінен қолданылуда.

1.3 Екілік кодтардың жіктелуі

Ариаға жіберілетін екілік кодтарды жіктеудің бірнеше критерийлері бар.

1-ші критерий – кодтардың атқаратын функциясы, бұл белгі бойынша арнаның кодтары қатені анықтайтын кодтар және қатені түзететін кодтар болып жіктеледі.

2-ші критерий – кодтардың жасалу тәсілі, бұл белгі бойынша кодтар блоктық кодтарға және жималаушы кодтарға бөлінеді.

3-ші критерий ақпараттық және тексеруші символдардың кодтық комбинацияда орналасуына, олардың комбинациядағы орындарының айқын көрінуіне негізделген, соған орай кодтар жүйеленген және жүйеленбеген болып жіктеледі.

Келесі классификациялау критерийі кодтық сөзді құрайтын әртүрлі символдардың санына негізделеді, соған сәйкес кодтар *екілік және екілік емес* болып бөлінеді. Практикада негізінен символдары 0 мен 1-ден, яғни екілік символдардан тұратын екілік

кодтар қолданылады. Осы оқу құралында екілік кодтар қарастырылады.

Енді осы екілік кодтардың жіктелуін қарастырайық. Екілік кодтар екі үлкен жеке топқа бөлінеді.

Бірінші топқа барлық болуы мүмкін комбинацияларды толық пайдаланатын кодтар жатады, олар **артықтығы жоқ** (немесе **артық емес, нензбыточные**) кодтар деп аталады. Әдебиеттерде оларды **қарапайым немесе бастапқы кодтар** деп те атайды.

Екінші топқа барлық болуы мүмкін комбинациялардың тек белгілі бір бөлігін ғана пайдаланылатын кодтар жатады. Мұндай кодтар 17657-72 МЕМСТ-қа сәйкес **артықтығы бар** (немесе **артық, избыточные**) кодтар деп аталады.

Артықтығы бар кодтар өз ішінен теңқалыпты (біркелкі) кодтар және теңқалыпты емес (әркелкі) кодтар болып екіге бөлінеді.

Ал теңқалыпты кодтар **үздіксіз кодтар мен блоктық кодтардан** тұрады.

Үздіксіз кодтарды тізбекті және жималаушы кодтар құрайды. Ал блоктық кодтар **бөлінбейтін кодтар және бөлінетін кодтардан** тұрады.

Бөлінбейтін кодтарға **Плоткин коды мен тұрақты салмағы бар кодтар** жатады. Бөлінетін кодтар тобы **жүйеленген кодтар мен жүйеленбеген кодтардан** тұрады. Жүйеленбеген кодқа **Бергер коды** кіреді.

Жүйеленген кодтарға келесі 11 код жатады:

1. Жұптылыққа бір тексеруі бар код.
2. Корреляциялық код.
3. Хэмминг коды.
4. Рид-Маллер коды.
5. Варшамов коды.
6. Итерациялық код.
7. Қарапайым қайталанатын код.
8. Инверсиялық код.
9. Голей коды.
10. Макдональд коды.
11. Жұптылықты аз тығыздықпен тексеру коды.

Жүйеленген кодтың келесі бір тобы — **циклдік кодтар**.
Циклдік кодтың 9 түрі бар:

1. $d=2$ қарапайым циклдік коды.

2. Хэмминг коды.
3. Боуз- Чоудхури-Хоквингем коды (БЧХ).
4. Мажоритарлы код.
5. Файер коды.
6. Абрамсон коды.
7. Милас-Абрамсон коды.
8. Рид-Соломон коды.
9. Компаундты код.

Артықтығы жоқ кодтар да артықтығы бар кодтар сияқты *теңқалыпты кодтар* және *теңқалыпты емес кодтар* болып екіге бөлінеді. Теңқалыпты емес кодтарға Шеннон-Фано коды мен Хаффман коды жатады.

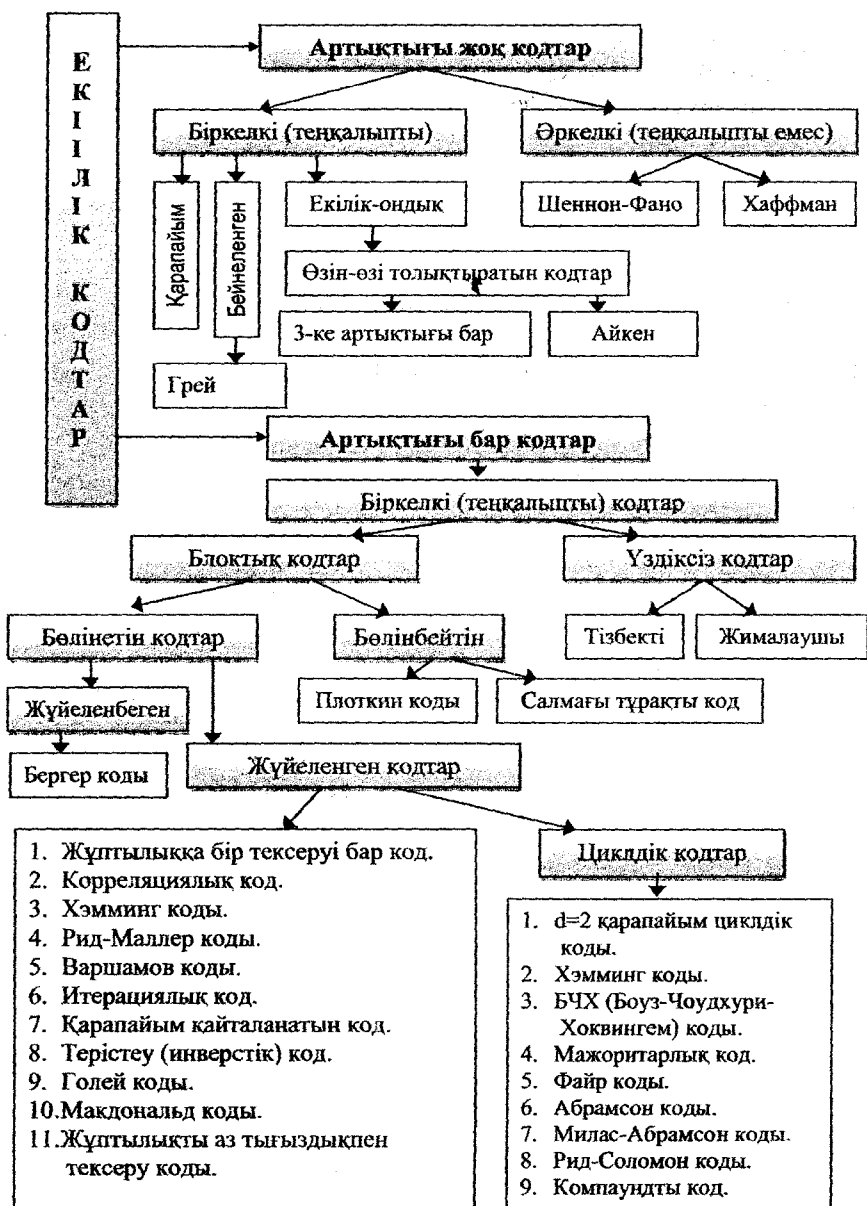
Теңқалыпты кодтар өзара үш топқа бөлінеді:

1) *екілік-ондық кодтар*, олардың ішінде өзін-өзі толықтыратын кодтарға 3 артықтығы бар екілік-ондық код (немесе 8-4-2-1 коды) және Айкен коды (немесе 2-4-2-1 коды) жатады.

2) *қарапайым кодтар*, оның мысалдарына халықаралық телеграф кодтары (МТК1, МТК2), КОИ-8 коды сияқты электронды есептеу машиналарына ақпаратты енгізу-шығару кодтары жатады.

3) *бейнеленген кодтар*, бұл кодтарда әрбір көрші 2 комбинацияның бір-бірінен тек 1 разрядқа ғана айырмашылығы болады. Мұндай кодқа Грей коды жатады.

Артықтығы жоқ теңқалыпты кодтар 1.4.2-бөлімде қарастырылады. Төмендегі 1.3-суретте жоғарыда айтылған екілік кодтардың жіктелуі схема түрінде көрсетілген.



1.3-сурет. Екілік кодтардың жіктелуі

1.4 Екілік кодтардың түрлері

1.4.1 Артықтығы бар кодтар

Алдыңғы бөлімдегі 1.3-суреттегі берілген екілік кодтардың жіктелуіне сәйкес артықтығы бар кодтарға сипаттама берейік.

Артықтығы бар (*орысша* избыточные) кодтар – жіберілетін ақпараттың барынша шынайылығын қамтамасыз етудің ең тиімді құралдарының бірі. Ақпаратты жіберу үшін артықтығы бар кодты құрастыру кезінде кодтық комбинациялардың кемінде бір разрядтағы бір-бірінен айырмашылығы бар бөлігі ғана (рұқсат етілген комбинация) пайдаланылады. Қалған комбинациялар рұқсат етілмеген комбинациялар болып саналады және пайдаланылмайды. Басқаша айтқанда, n кодтық комбинацияның ішінен хабар жіберу үшін тек k символ пайдаланылады, олай болса, $N=2^n$ мүмкін болатын барлық кодтық комбинациялардың ішінен тек $N_x=2^k$ рұқсат етілген комбинациялар қолданылады. Артықтығы бар кодтарды қолданғанда бір разрядтағы қате рұқсат етілген комбинацияны рұқсат етілмеген комбинацияға алмастырады. Анықталатын қате комбинациялардың үлесі [1]:

$$N_x(N - N_x) / N_x N = 1 - N_x / N = 1 - 2^k / 2^n.$$

Түзетілетін қате комбинациялардың үлесі:

$$(N - N_x) / N_x(N - N_x) = 1 / N_x.$$

Артықтығы бар және артықтығы жоқ кодтардың екі тобы да өз кезегінде тең қалыпты (біркелкі) және тең қалыпты емес (әркелкі) болып бөлінеді. Тең қалыпты кодтар – бұл барлық кодтық комбинацияларының құрамындағы разрядтар саны тұрақты болатын кодтар. Тең қалыпты емес кодтар – кодтық комбинацияларының құрамында разрядтар саны әртүрлі болатын кодтар. Артықтығы бар кодтар техникалық жүзеге асырылуының қиындығына байланысты іс жүзінде қолданылмайды.

Барлық артықтығы бар кодтар үздіксіз кодтар және блоктық кодтар болып екі топқа бөлінеді.

Үздіксіз кодтар. Үздіксіз (*орысша* непрерывные) кодтарда кодтау және декодтау процесі үздіксіз сипатта болады, кодтау және декодтау операциялары ақпараттық символдардың тізбегіне оларды блоктарға бөлмей үздіксіз жүргізіледі. Үздіксіз кодтар **тізбекті код және жималаушы кодтар** болып бөлінеді. Бұл кодтар қателер тобын анықтау және түзету үшін қолданылады.

Тізбекті код (*орысша* цепной код). Бұл кодта әрбір ақпараттық элементтен кейін тексеруші элемент келеді. Тексеруші элементтер екілік модуль бойынша k қосу кадамына бір-бірінен қалып отырған екі ақпараттық элементтерді қосу арқылы жасалады. Қосу кадамы k – бұл тексеру элементін құрайтын екі ақпараттық элемент арасындағы арақашықтық. Кодтың түзету қабілеті қосу кадамына тәуелді болғандықтан, оны өзгерте отырып, ұзындығы $2k$ қателер тобын анықтауға және түзетуге арналған кодты жасауға болады. Бұл кодтың артықтығы 0,5 тең. Тізбектік код салыстырмалы түрде алғанда қарапайым, алайда үлкен артықтығы болғандықтан байланыс арналарында туындайтын қателерді (қателер тобын) табуға немесе түзетуге мүмкіндік береді. Қосу кадамының шамасын өзгерте отырып, кодтың түзету мүмкіндіктерін байланыс арналарының сипаттамасымен келістіруге, яғни қателердің болуы ықтимал жиілігін артыруға немесе азайтуға болады.

Мысал. $a_1, a_2, a_3, a_4, a_5, \dots$ символдар тізбегі берілсін. Қосу кадамы $L = 2$. Ақпараттық символдарды екілік модуль бойынша қосып, тексеруші символдарды құрамыз: $b_{13} = a_1 + a_3, b_{24} = a_2 + a_4 \dots$ Алынған тексеруші символдарды ақпараттық символдардың арасына λ ығысумен орналастырады. Осы ығысу кодтық тізбектегі ақпараттық және тексеруші символдардың арасындағы корреляциялық байланысты көрсетеді. Нәтижесінде мұндай кодпен бір қатені, екі қатені және қателер тобын да түзету мүмкін болады.

Тізбектік кодтың декодталуы: қабылданған хабардан ақпараттық элементтер мен тексеруші элементтер жеке бөлініп алынады. Бөлініп алынған ақпараттық разрядтардан тізбектік кодты құру ережесіне сәйкес тексеруші элементтер есептелініп, құрастырылады. Әрбір құрастырылған тексеруші элемент қабылдан алынған тексеруші элементпен екілік модуль бойынша салыстырылады. Қате болмаған жағдайда екеуінің разрядтары

бірдей болады. Бірдей емес разрядтар қатенің бар болуын білдіреді.

Жималаушы кодтар (*орысша* сверточные коды) оқу құралының 5-ші бөлімінде қарастырылады.

Блоктық кодтар. Блоктық кодтарда әрбір хабарға ұзындығы n символдан тұратын кодтық комбинация (блок) сәйкес келеді. Блоктар бір-біріне тәуелсіз жеке кодталады және декодталады. Сызықтық блоктық кодтар оқу құралының 4.2-ші бөлімінде қарастырылады.

Блоктық кодтар *бөлінетін және бөлінбейтін кодтар* болып екіге бөлінеді.

Бөлінбейтін кодтар. Бұл топқа жататын кодтар саны аз. Оларға *салмағы тұрақты кодтар* мен *Плоткин кодтары* жатады.

Бөлінбейтін кодтарда кодтық комбинацияның (кодтық вектор, кодтық сөз деп те атайды) құрамы ақпараттық және тексеруші символдар болып нақты бөлінбейді.

Салмағы тұрақты код – қарапайым блоктық код, практикада кеңінен қолданылады, мысалы 7-ден 3 коды (7 разрядты комбинацияларда бірдің саны 3). Салмағы тұрақты кодта кодтық комбинация құрамын (блокты) ақпараттық және тексеруші символдарға бөлуге болмайды. Бұл кодтың барлық комбинацияларының салмағы бірдей, яғни тұрақты болады (комбинациялардағы нөлдердің, бірлердің саны бірдей). Бірлердің санының өзгеруі қатенің бар екенін білдіреді. Бұл кодтарды қолданудың негізгі себептерінің бірі – айтарлықтай дәрежеде асимметриялы байланыс арналарын пайдаланған кездегі олардың артықшылығы. Толық асимметриялық арна жағдайында тұрақты салмағы бар код барлық қателерді табатындықтан, қателерді табуға қатысты жетік код деп саналады. Тек нөлдердің бірлерге, бірлердің нөлдерге өзгеруі ғана орын алған арна толық асимметриялық болып табылады. Екілік симметриялық арнада (қателердің екі түрі де бірдей болатын арна) салмағы тұрақты кодтар қателердің барлық болуы мүмкін шексіз санын табады, олардың ішінде нөлдің бір позицияда бірге өзгеруі бірдің басқа позициядағы нөлге өзгеруімен орны толады. Кодталған сөздің белгіленген ұзындығында салмағы тұрақты оңтайлы кодтың әдетте оған ұқсас анықталған сипаттамасы бар басқа бөлінетін кодқа қарағанда рұқсат берілген

комбинациялары көбірек болады. *Салмағы тұрақты кодтың негізгі кемшілігі* – оның құрамындағы ақпараттық және тексеруші разрядтардың айқын түрде бөлінбейтіндігінде.

Салмағы тұрақты кодтардағы анықталған мүмкіндік кодтық сөздің жазылуымен түсіндіріледі, осыған орай кодтың өзгеріссіз тексеру белгілерін ажырату мүмкін емес. Осылайша бөлінбеуге байланысты анықталған кодтың мүмкіндігі – бірдің нөлге өзгеру мүмкіндігінің нөлдің бірге өзгеру мүмкіндігіне тең болуында. Салмағы тұрақты код тек қателерді анықтайтын жүйелерде ғана пайдаланылады. Бұл код негізінде міндетті түрде үнемі дискреттік нысанға өзгертуді қажет ететін символдық-сандық ақпаратты жіберу үшін қолданылады. Ол кез-келген дара қателерді, жиі кездесетін қателердің біразын анықтайды. Бір мезгілде $0 \rightarrow 1$, $1 \rightarrow 0$ -ге ауысатын ығысу қателерін таба алмайды.

Мысал. C_5^2 және C_7^3 салмағы тұрақты кодтары:

| $C_5^2 = N=10$ (комбинация саны) n=5 разрядтан тұрады, салмағы w=2 (1 саны) | $C_7^3 = N=35$ (комбинация саны) n=7 разрядтан тұрады, салмағы w=3 (1 саны) |
|---|---|
| 11000 | 0000111 |
| 10010 | 1001001 |
| 00101 | 1010100 |
| | |

Плоткин коды. Моррис Плоткин (АҚШ) кодын Адамар матрицаларынан алады. Кодтың ерекшелігі – бұл код тиімділігі жағынан Плоткиннің теориялық шегіне жетеді. Адамар матрицасы – элементтері 1 және -1 болатын, өлшемі $n \times n$ ортогональды квадраттық матрица.

$$2 \times 2 \text{ матрица } H_2 = \begin{vmatrix} 1 & 1 \\ 1 & -1 \end{vmatrix}$$

$$4 \times 4 \text{ матрица } H_4 = \begin{vmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{vmatrix}$$

Мұндай матрица арқылы кодтық қашықтығы үлкен кодтарды құруға болады. Плоткин кодының түзету қабілеті жоғары, алайда кодтау мен декодтаудың практикалық әдістері болмағандықтан, қолданбалы жағынан маңызы аз.

Плоткиннің теориялық шегі кодтау теориясында ұзындығы n және минимальды қашықтығы d болатын екілік кодтың қуатын, яғни $A_2(n, d)$ кодтық сөздердің максималды мүмкін болатын санын анықтайды.

$$d = \min_{\substack{x, y \in C, \\ x \neq y}} d(x, y),$$

Мұндағы $d(x, y)$ – бұл x, y кодтық векторларының арасындағы Хэмминг қашықтығы (Хэмминг қашықтығы 4.2.5-бөлімде қарастырылады).

- 1) егер d – жұп сан және $2d > n$ болса, онда

$$A_2(n, d) \leq 2 \left\lfloor \frac{d}{2d - n} \right\rfloor.$$

- 2) егер d – тақ сан және $2d + 1 > n$ болса, онда

$$A_2(n, d) \leq 2 \left\lfloor \frac{d + 1}{2d + 1 - n} \right\rfloor.$$

- 3) егер d – тақ сан болса, онда

4)

$$A_2(2d + 1, d) \leq 4d + 4,$$

- 5) егер d – жұп сан болса, онда

6)

$$A_2(2d, d) \leq 4d.$$

Ескерту. $\lfloor \rfloor$ операторы санның бүтін бөлігін білдіреді.

Плоткин коды туралы толығырақ мына әдебиеттен [Плоткин М. Двоичные коды с заданным минимальным расстоянием. - Кибернетический сборник. - Москва, 7:60-73 [2], 1963] танысуға болады.

Бөлінетін кодтар. Бөлінетін кодтар деп блоктың (немесе үздіксіз тізбектің) құрамындағы символдардың ролі ақпараттық және тексеруші символдар болып нақты шектелген кодтарды айтады. Ақпараттық символдарға тіркесіп жазылатын тексеруші символдар қателерді анықтау немесе түзету үшін қолданылады. Ақпараттық және тексеруші символдар барлық кодтық векторларда бірдей позицияларда орналасады. кодтық векторлар саны – $2k$.

Бөлінетін блоктық кодтар әдетте (n, k) кодтары деп белгіленеді, мұндағы n – кодтық комбинациядағы разрядтар саны, k – кодтық комбинациядағы ақпараттық символдар саны.

Бөлінетін блоктық кодтар өз кезегінде *жүйеленген* (орысша систематические) және *жүйеленбеген* (несистематические) болып келеді.

Жүйеленбеген кодтар. Жүйеленбеген кодтардағы ақпараттық символдардың тізбегі ұзындығы 1-ге тең ішкі блоктарға бөлініп, тексеруші символдар (разрядтар) осы ішкі блоктардың қосындысынан тұрады. Мұндай кодтар серия ұзындығы 1-ден аспайтын сериялық қателерді анықтауға мүмкіндік береді. Жүйеленбеген кодтарға Бергер коды (немесе қосуы бар кодтар) жатады.

Бергер коды. Бергер кодын құрастырудың бірнеше нұсқасы бар. Ең қарапайым нұсқада кодтау былайша жүргізіледі: кодтың ақпараттық бөліміндегі бірліктер саны есептеледі, содан соң осы санның екілік түрдегі терістелген жазуы болып табылатын тексеруші разрядтар қалыптасады. Бұдан, R тексеруші разрядтарының саны $\log_2(k)$ -дан артық ең аз бүтін санға тең, яғни $R \geq \log_2(k)$.

Мысал. 011010 хабары Бергер кодымен кодталғанда 011010100 болады. 011010 хабарындағы 1-дің саны үшеу, екілік кодта 3 санының жазылуы 011, оны терістегенде 100 болады, осы сан тексеруші символдар болады.

Бергер кодтары тек нөлдерді (0) бірліктерге (1) немесе керісінше өзгертуге болатын асимметриялық байланыс арнасында ғана пайдалануға арналған.

Салмағы тұрақты кодтармен салыстырғанда Бергер кодтарының артықшылығы – олар тексеруші символдар бөлігін құрастырудың өте қарапайым алгоритмі бар бөлінетін кодтар болып

табылады. Симметриялық арналарда мұндай кодтар дара кездесетін қателердің барлығын және бірнеше қателердің кейбір бөлігін ғана табады. Симметриялық арналар үшін тез тауып алу қасиеттері бар кодтарды құрастыруға болады. Мұндай кодтарда әрбір ақпараттық позицияға әртүрлі салмақ беріледі, бұл ретте бірде бір салмақ екінің дәрежесі болмайды (3, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 17 және т.с.с.). Мұның тексеру белгілері 1 орналасқан ақпараттық разрядтарға сәйкес келетін салмақтарды қосу жолымен құрастырылады. Содан соң алынған сан терістеледі. Мұндай код симметриялық арнада қос қателерді табады және дара қателерді тауып, түзетеді. Тексеру символдарының саны мынадай арақатынаспен анықталады:

$$r \geq \text{Log}_2((k + m) \cdot (k + m + 1) / 2 - 2 \cdot m + 1),$$

мұндағы k – ақпараттық символдардың саны, ал m – қос теңсіздіктен анықталады: $2m-1 < k + m < 2m$.

Жүйеленген кодтар. Бөлінетін блоктық кодтардың ең үлкен класын жүйеленген кодтар немесе *топтық* болып табылатын *сызықтық кодтар* құрайды. «Топтық» деп аталу себебі код екілік модуль ($\text{mod } 2$) бойынша қосу операциясына топ болады, яғни екі кодтық вектордың (кодтық сөздің, кодтық комбинацияның) $\text{mod } 2$ бойынша қосындысы кодтық вектор болады.

Жүйеленген кодтардың басты ерекшелігі – олардың тексеруші символдары ақпараттық символдардың әртүрлі сызықтық комбинациялары болады. Жүйеленген кодтарды құру әдісі туындатушы (жасаушы, құрушы деп те атайды) матрицаны қолдануға негізделген, бұл матрицаның жолдары сызықтық тәуелсіз базистік векторлар болып табылады. Жүйеленген кодтардың барлық кодтық векторлары туындатушы матрица жолдарының екілік модуль бойынша сызықты комбинациялары ретінде жасалады.

Жүйеленген кодтарға жұптылыққа бір тексеруші бар кодтар, қайталанатын код, корреляциялық код, терістеу (инверстік) коды, Хэмминг, Голей, Рид-Маллер, Макдональд, Варшамов кодтары, жұптылыққа аз тығыздықпен тексеру коды, итерациялық код жатады. Аталған кодтар сызықты блокты кодтарға арналған 4.2-бөлімде қарастырылады.

Жүйеленген кодтардың келесі бір үлкен класына циклдік кодтар жатады. Циклдық кодтар 4.3-бөлімде қарастырылады.

1.4.2 Артықтығы жоқ кодтар

Артықтығы жоқ кодтарда мүмкін болатын барлық комбинациялар толық пайдаланылады. Артықтығы жоқ кодтарға ең аз кодтық арақашықтығы $d_{\min}=1$ болатын комбинацияларды ішінара пайдаланатын кодтар да жатады. Мұндай кодтар тіпті бір реттік бөгеуіден туындаған қателердің де барлық түрлерін таппайды.

n -разрядты тізбектен алуға болатын әртүрлі комбинациялардың саны мына 1.3-формула бойынша айқындалады :

$$N=C_n^0+C_n^1+C_n^2+\dots+C_n^n \quad (1.3)$$

мұндағы $C_n^0, C_n^1, \dots, C_n^n$ бойынша n элементтер қосындысының саны. Тиісті өзгерістерден кейін $N=2^n$ аламыз. Себебі, мұндай жағдайда барлық болуы мүмкін комбинациялар пайдаланылады, n – символдардың болуы мүмкін мәндерінің санында 2 (яғни 0 мен 1) олардың көмегімен кодтық комбинациялардың барынша көп санын құрастыруға болатын разрядтардың ең аз саны. Осыған орай N ақпаратты беру үшін қажетті разрядтардың саны мынаған тең: $n=\log_2 N$.

Артық емес кодтар теңқалыпты (біркелкі) және теңқалыпты емес (эркелкі) кодтарға бөлінеді.

Теңқалыпты кодтар. Теңқалыпты (*орысша* равномерные) кодтар өз ішінен екілік-ондық кодтар, қарапайым кодтар және бейнеленетін кодтар болып 3 топқа бөлінеді.

Екілік-ондық кодтар (17 нұсқасы бар). Бұл кодтауда әрбір ондық сан (хабар) 4 элементтен тұратын екілік символдардан құрастырылады. Екілік кодта жазылған 4-разрядты барлық мүмкін болатын комбинациялардың жалпы саны $N=2^4=16$, бұдан ондық жүйеде ондық санды бейнелеу үшін тек 10 комбинация пайдаланылады, ал қалған 6 комбинация артық болып табылады., бұл кодтың көптеген нұсқаларын құруға мүмкіндік береді. Ондық цифрларды екілік түрде көрсету үшін 2 цифрдың бастапқы 4 дәрежесін $2^0, 2^1, 2^2, 2^3$ ($2^0=1; 2^1=2; 2^2=4; 2^3=8$) қолданғанда 8-

4-2-1 ($q_4-q_3-q_2-q_1$) кодын аламыз. Бұл кодтың әрбір разрядының салмағы тұрақты.

Екілік-ондық кодты ($q_4-q_3-q_2-q_1$) құру үшін келесі шарттар орындалуы тиіс:

- 1) Ең аз мәні бар цифр q_1 -дің салмағы 1-ге тең.
- 2) Ең аз мәні бар екінші цифр q_2 -нің салмағы 1 немесе 2 болады.
- 3) Қалған екі код сандарына сәйкес келетін салмақты олардың қосындысы 6-дан (егер $q_2 = 2$) немесе 7-ден (егер $q_2 = 1$) үлкен немесе оған тең болатындай етіп тандайды.

Осыған байланысты кодтардың 17 түрін алуға болады. Екілік-ондық кодтардың 8-4-2-1 басқа төмендегідей 16 нұсқалары бар:

| | | | |
|---------|---------|---------|---------|
| 7-4-2-1 | 7-3-2-1 | 3-3-2-1 | 6-3-1-1 |
| 6-4-2-1 | 6-3-2-1 | 6-2-2-1 | 5-3-1-1 |
| 5-4-2-1 | 5-3-2-1 | 5-2-2-1 | 4-3-1-1 |
| 4-4-2-1 | 4-3-2-1 | 4-2-2-1 | 5-2-1-1 |

Бұл кодтар 0-9 аралығындағы ондық санды бірмәнді түрде бейнелемейді.

Мысал. 4-3-2-1 коды 6 ондық санын екі түрде бейнелейді: 6 санына 2 комбинацияны 0111 немесе 1010 сәйкестендіреді. Ондық-екілік кодтар дискретті телеөлшеуіш жүйелерінде өлшенетін параметрлер цифрлық индикаторларда көрсетілетін жағдайда қолданылады.

Кодтау ережесі. Бір ондық цифрды екілік-ондық кодқа ауыстыру үшін екілік-ондық кодтың 1-ші үлкен екілік разрядынан бастап ағымдағы (екілік) цифр салмағының қалдық саннан артық болуын тексеру керек; егер аз немесе тең болса, онда шығу кодына 1 цифрын жазады және қалдық саннан екілік-ондық кодтағы екілік цифр салмағын азайту керек. Ағымдағы қалдықтың бастапқы мәні кодталатын санмен бірдей болады. Ал ағымдағы (екілік) цифр салмағы қалдық саннан артық болса, онда шығу кодына 0 жазады.

Декодтау ережесі. Екілік-ондық кодта жазылған хабарды декодтау үшін мәні 1-ге тең разрядтар салмағының мәнін қосу керек. Екілік-ондық 6-4-2-1 кодымен кодталған 0101 хабарын декодтау нәтижесі: $0 \cdot 6 + 1 \cdot 4 + 0 \cdot 2 + 1 \cdot 1 = 5$.

Кодтау мысалдары:

- а) 5 санын 6-4-2-1 кодымен кодтау.

1) 6-4-2-1 кодындағы 1-ші бит салмағы – 6, ағымдағы қалдықтан артық (ағымдағы қалдықтың бастапқы мәні кодталатын 5 санымен бірдей болады), бұдан екілік-ондық кодтың 1-ші цифры – 0.

2) 6-4-2-1 кодындағы 2-ші бит салмағы – $4 < 5$. Жасалатын кодтың келесі биті 1-ге тең және ағымдағы қалдық $5 - 4 = 1$ -ге тең.

3) 6-4-2-1 кодындағы 3-ші бит салмағы – 2, ағымдағы қалдықтан (1) артық. Жасалатын кодтың келесі биті 0 болады.

4) 4-ші бит салмағы – 1, қалдық санға (1) тең. Шығу кодының соңғы саны – 1. Соңғы қалдығы $1 - 1 = 0$. Оның 0-ге теңдігі кодтың дұрыстығын білдіреді.

5 санын 6-4-2-1 кодымен кодтау нәтижесі – 0101 болады.

б) 801 санын 5-2-1-1 екілік-ондық кодымен кодтау нәтижесі 1110 0000 0010 болады.

$8 = 5 \cdot 1 + 2 \cdot 1 + 1 \cdot 1 + 1 \cdot 0$; (8 санына 1110 сәйкес келеді)

$0 = 5 \cdot 0 + 2 \cdot 0 + 1 \cdot 0 + 1 \cdot 0$; (0 санына 0000 сәйкес келеді)

$1 = 5 \cdot 0 + 2 \cdot 0 + 1 \cdot 1 + 1 \cdot 0$. (1 санына 0010 сәйкес келеді)

с) 927 саны 7-4-2-1 екілік-ондық кодымен кодтау нәтижесі 1010 0010 1000 болады:

$9 = 7 \cdot 1 + 4 \cdot 0 + 2 \cdot 1 + 1 \cdot 0$;

$2 = 7 \cdot 0 + 4 \cdot 0 + 2 \cdot 1 + 1 \cdot 0$;

$7 = 7 \cdot 1 + 4 \cdot 0 + 2 \cdot 0 + 1 \cdot 0$.

Декодтау мысалдары:

а) 6-3-2-1 екілік-ондық кодымен кодталған 001001011100 хабарын декодтау нәтижесі 249 болады:

$6 \cdot 0 + 3 \cdot 0 + 2 \cdot 1 + 1 \cdot 0 = 2$;

$6 \cdot 0 + 3 \cdot 1 + 2 \cdot 0 + 1 \cdot 1 = 4$;

$6 \cdot 1 + 3 \cdot 1 + 2 \cdot 0 + 1 \cdot 0 = 9$.

б) 8-4-2-1 екілік-ондық кодымен кодталған 100101110010 хабарын декодтау нәтижесінде 972 саны шығады.

$8 \cdot 1 + 4 \cdot 0 + 2 \cdot 0 + 1 \cdot 1 = 9$

$8 \cdot 1 + 4 \cdot 1 + 2 \cdot 1 + 1 \cdot 1 = 7$

$8 \cdot 0 + 4 \cdot 0 + 2 \cdot 1 + 1 \cdot 0 = 2$

Өзін-өзі толықтыратын кодтар. Есептеу техникасында екілік-ондық кодтардың өзін-өзі 9-ға дейін толықтыру қасиеті бар түрі – *өзін-өзі толықтыратын кодтар* кеңінен қолданылады. Бұл кодтар ЭЕМ-да азайту операциясын қосу операциясымен алмастыру үшін арнайы машиналық кодтарда (кері және толықтырушы кодтарда) қолданылады. Кері кодта әрбір $[a_i]_2$ разряд

2) 1-ге дейін толықтыру ретінде беріледі, яғни $[a_i]_2 = 2 - 1 - g_i$.
 Егер $a_i = 1$ болса, онда $2 - 1 = 1$ -ге дейінгі толықтыру 0-ге тең, ал егер $a_i = 0$ болса, онда толықтыру 1-ге тең, яғни a_i санының инверсиясы (терістеуі) болып табылады. Ондық код үшін 9-ға дейін толықтыруды табу қажет. Өздігінен толықтырылатын екілік-ондық кодтардың ыңғайлы болу себебі мынада: 9-ға дейін толықтыру болатын екілік-ондық кодтың цифры кодтағы толықтыру іделінетін ондық санның екілік түрін терістеу (инверсия) арқылы табылады. Сондықтан егер a_i ондық санның разрядын g_4, g_3, g_2, g_1 түріндегі төрт екілік символдар тобы арқылы бейнелесек, онда 9-ға дейінгі толықтыру $[a_i]_{10} = \overline{g_4 g_3 g_2 g_1}$ ретінде беріледі, мұндағы $\overline{g_i}$ — бұл g_i екілік санының терістеуі.

Айкен коды және 3-ке артықтығы бар код. Өзін-өзі толықтыратын кодтардың ішінде ең кеңінен тарағаны — *Айкен коды* деп аталатын **2-4-2-1 код** және *3-ке артықтығы бар код* деп аталатын **8-4-2-1 коды**. Төмендегі 1.2-кестеден кодтың барлық төрт разрядтарының цифрларын 0-ден 1-ге (немесе керісінше) ауыстырғанда кодталатын ондық цифр үшін 9-ға дейін толықтыру шығатыны көрініп тұр.

Кесте 1.2 — Айкен коды және 3-ке артықтығы бар код

| Ондық сан | Айкен коды (2-4-2-1) | 3-ке артықтығы бар код (8-4-2-1) |
|-----------|----------------------|----------------------------------|
| 0 | 0000 | 0011 |
| 1 | 0001 | 0100 |
| 2 | 0010 | 0101 |
| 3 | 0011 | 0110 |
| 4 | 0100 | 0111 |
| 5 | 1011 | 1000 |
| 6 | 1100 | 1001 |
| 7 | 1101 | 1010 |
| 8 | 1110 | 1011 |
| 9 | 1111 | 1100 |

Қарапайым кодтар. Қарапайым кодтарда (артықтығы жоқ кодтар деп те атайды) хабарды жіберу үшін мүмкін болатын барлық комбинациялар толық пайдаланылады. Қарапайым екілік кодтың сыйымдылығы, яғни барлық мүмкін болатын комбинациялардың жалпы саны $N = 2^n$ болады.

Мысал. 3 разрядты $n=3$ екілік код үшін $N = 2^3=8$, бұл кодтың барлық комбинациялары төмендегі 1.3-кестеде көрсетілген.

Кесте –1.3. Үш разрядты екілік код комбинациялары

| Комбинация № | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|--------------|-----|-----|-----|-----|-----|-----|-----|-----|
| Екілік код | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |

Егер S хабар жіберу қажет болса, онда қарапайым екілік кодтағы разряд саны мынаған тең болу керек: $n = \lceil \log_2 S \rceil$

Мұндағы, $a = \log_2 S$ деп белгілесек, $\lceil a \rceil$ – бұл a мәніне жақын бүтін санды білдіреді, $A > a$. Егер $S = 50$ болса, онда $n = \lceil \log_2 50 \rceil = 6$.

Бұл формула S хабарды жіберуге қажетті кодтың минимальды элементтер санын анықтайды. Қарапайым кодтың **артықтығы** (*орысша избыточность*) мынаған тең (1.4-формула):

$$R = \frac{n}{\lceil \log_2 S \rceil} = \frac{n}{\lceil \log_2 2^n \rceil} = 1 \quad (1.4)$$

Қарапайым кодтың кемшілігі: хабарды жіберу барысындағы қатенің пайда болу ықтималдығы кодтың ұзындығы артқан сайын сызықты түрде артады, яғни көп ақпарат жіберілген сайын оның сенімділігі төмендейді.

Қарапайым екілік код элементтің бүліну ықтималдығы аз болған жағдайда кеңінен қолданылады, өйткені барлық кодтардың ішінде оның сыйымдылығы ең үлкен, әрі оны кодтау және декодтау үшін қарапайым аппаратура пайдаланылады.

Бейнеленген кодтар (*орысшасы отраженные; шағылысқан, бейнеленетін* деп те аударылып жүр). Екілік кодта бір санның бейнелеуінен көршілес үлкен немесе көршілес кіші санды бейнелеуге көшкенде бірнеше разрядтардағы сандар бір уақытта өзгереді. *Мысалы,* 7 санынан $(0111)_2$ 8 санын $(1000)_2$ бейнелеуге өткен кезде бір уақытта барлық төрт разрядтағы сандар өзгереді. Бұл үздіксіз хабарды екілік кодтарға кодтаудың кейбір тәсілдерінде айтарлықтай қателер туындатуы мүмкін. Бір мәнді емес қатемен күресудің жиі қолданылатын, әрі тиімді құралдарының бірі – бұл арнайы кодтарды, яғни бейнеленген (рефлексті, *ағылш. to reflect* – бейнелеу) кодтарды қолдану. *Бейнеленген кодтардың басты*

ерекшелігі – кодтарда әрбір көрші 2 комбинацияның бір-бірінен тек 1 разрядқа ғана айырмашылығы болады. Осылайша, бұл кодтарда бірнеше разрядтардағы сандар бір уақытта өзгермейді.

Екі көршілес кодтар комбинацияның бір-бірінен тек бір символға ғана айырмашылығы болатындай етіп екілік кодтардың көп санын құрастыруға болады. Төмендегі 1.4-кестеде бейнеленетін кодтардың нұсқалары берілген.

Кесте 1.4 – Бейнеленген кодтар нұсқалары

| Ондық сан | Бейнеленген кодтар | | | | | | |
|-----------|--------------------|---------|---------|---------|---------|---------|---------|
| | 1 нұсқа | 2 нұсқа | 3 нұсқа | 4 нұсқа | 5 нұсқа | 6 нұсқа | 7 нұсқа |
| 0 | 000 | 000 | 000 | 000 | 000 | 000 | 000 |
| 1 | 010 | 100 | 100 | 001 | 001 | 001 | 001 |
| 2 | 011 | 101 | 110 | 011 | 101 | 101 | 101 |
| 3 | 001 | 001 | 010 | 010 | 111 | 100 | 100 |
| 4 | 101 | 011 | 011 | 110 | 011 | 110 | 110 |
| 5 | 111 | 111 | 111 | 111 | 010 | 111 | 111 |
| 6 | 110 | 110 | 101 | 101 | 110 | 011 | 011 |
| 7 | 100 | 010 | 001 | 100 | 100 | 010 | 010 |

Бейнеленген кодтардың ішінде жиі қолданылатыны – Грей коды.

Грей коды. Грей коды бөгеуілдерге орнықты етіп кодтайтын, қатені табатын кодтарға жатады. Грей кодының екілік кодқа түрлендіру әдістері қарапайым, ең жиі қолданылатын қатені анықтайтын кодтардың бірі болып саналады.

Төмендегі 1.5-кестеде төрт разрядты қарапайым екілік кодтың және Грей кодының комбинациялары көрсетілген.

Кесте 1.5 – Қарапайым екілік код және Грей коды

| Ондық сан | Қарапайым екілік код | Грей коды |
|-----------|----------------------|-----------|
| 0 | 0000 | 0001 |
| 1 | 0001 | 0011 |
| 2 | 0010 | 0010 |
| 3 | 0011 | 0110 |
| 4 | 0100 | 0111 |
| 5 | 0101 | 0101 |

| | | |
|----|------|------|
| 6 | 0110 | 0100 |
| 7 | 0111 | 1100 |
| 8 | 1000 | 1101 |
| 9 | 1001 | 1111 |
| 10 | 1010 | 1110 |
| 11 | 1011 | 1010 |
| 12 | 1100 | 1011 |
| 13 | 1101 | 1001 |
| 14 | 1110 | 1000 |
| 15 | 1111 | 0001 |

Грей коды айналып тұрған дененің бұрылу бұрышын кодқа түрлендіру үшін қолданылады. Білікте айналып тұрған пластинада жарық өтетін саңылаулар болады. Диск секторларға бөлінеді, саңылаулар осы секторларда жасалынады. Дисктің айналу барысында саңылаулардан өткен жарық фотоқабылдағышты іске қосады. Ақпаратты екілік код түрінде түсірген кезде елеулі қате болуы мүмкін. Мысалы, 7 мен 8 цифрларын алайық. Бұл цифрлардың екілік кодтары барлық разрядтарда өзгеше.

7 0111 → 1111
8 1000 → 0000

Егер қате жоғарғы разрядта кездессе, онда өте үлкен қате (360°) болады. Ал Грей кодында көрші комбинациялардың бір-бірінен тек 1 символға ғана айырмашылығы бар, сондықтан бір санды бейнелеуден көрші екіншісіне ауысқанда тек кіші разрядтың бірлігі өзгеріп, қате өте аз болады.

Грей кодының ерекшеліктері. Грей кодының әрбір келесі комбинациясы алдыңғысынан тек бір разрядқа ғана өзгеше болады. Комбинациялар арасында ауысқанда қарапайым екілік кодқа қарағанда Грей кодында әр разрядтағы мәндердің өзгеруі 2 есе сирек болады, кодтаудың жоғары дәлдігін береді. Екі көрші комбинацияны екілік модуль бойынша қосқанда шығатын бірлердің саны разрядтар санынан 3-ті азайтқанға тең. Ұзындығы n -ге тең код үшін бірлердің саны (код салмағы) $n-1$ болады. Грей кодының осы қасиеті қабылданған комбинациялардың дұрыстығын тексеру үшін қолданылады. Грей кодында кейбір разрядтардың бірдей екендігін көрсететін симметрия осін

(«бейнелену» осін) байқауға болады. Мысалы, 7 мен 8 сандарының арасында жүргізілген оське қатысты симметрия болады. Ұзындығы n -ге тең код Грей кодында $2^{n-1}-1$ және 2^{n-1} сандары арасында өтетін симметрия осі симметрияның басты осі деп аталады, осы басты оське қатысты симметриялы кодтық комбинациялардың $n-1$ разрядтарындағы символдар бірдей болады. Осы ерекшелік «бейнеленген (рефлексті) код» терминінің енгізілуіне себепкер болды. Қарапайым екілік кодпен салыстырғанда бізмәнді емес түрде оқу қателерін және байланыс арнасындағы бөгеуілдерден туындаған қателерді азайтады. Грей коды әртүрлі үздіксіз хабарларды аналогтік-цифрлық түрлендіру үшін қолданылады.

Грей кодының кемшіліктері. Басқа да бейнеленетін кодтардағы сияқты басты кемшілігіне олардың салмақсыздығы, яғни бірдің салмағының разрядтың номері арқылы анықталмауы жатады. Мұндай кодтармен берілген ақпаратты ЭЕМ-да өңдеу қиын, декодтаушы құрылғылардың күрделі болуына байланысты декодтауы да қиын. Сондықтан да ЭЕМ-ге енгізу алдында бейнеленген кодтарды қарапайым екілік кодқа айналдырады, ал олар оңай декодталады.

Қарапайым екілік кодты Грей кодына аудару үшін екілік санның астына сол санның өзі оңға бір разрядқа жылжытып жазылады, соңғы разряд жоғалады. Одан кейін екеуінің разрядтарын екілік модуль бойынша қосады, нәтижесінде Грей коды жасалады.

(Ескерту. Екілік модуль бойынша қосу ережесі: $1\oplus 1=0$, $0\oplus 0=0$, $1\oplus 0=1$, $0\oplus 1=1$
 Бірдей разрядтар қосындысы 0, әртүрлі разрядтар қосындысы 1 болады)

$$\begin{array}{l} 11010011_2 \\ \oplus \underline{1101001}_2 \text{ (оңға 1 орынға жылжытқанда соңғы 1 жазылмайды)} \\ 10111010_{\text{Грей}} \quad 11010011_2 = 10111010_{\text{Грей}} \end{array}$$

Грей кодын қарапайым екілік кодқа аудару үшін 1-ші разряд цифры өзгеріссіз жазылады. Әрбір келесі цифр өзінің алдында неше 1 цифр тұрса, сонша рет терістеледі.

$$1010_{\text{Грей}} = 1100_2$$

Артықтығы жоқ теңқалыпты емес кодтар. Екілік теңқалыпты (біркелкі) кодпен кодталған хабарларды жібергенде әдетте жіберілетін хабардың статистикалық құрылымы ескерілмейді. Пайда болуы мүмкіндігіне қарамастан, барлық хабар ұзындығы бірдей кодтық комбинациялар болып табылады, яғни бір хабарға келетін екілік символдардың саны қатаң түрде тұрақты келеді.

Артықтығы жоқ теңқалыпты емес кодтарда барлық мүмкін болатын комбинациялардың барлығы пайдаланылады және кодтық комбинациялардың ұзындықтары, яғни олардағы разрядтар саны әркелкі болады. Артықтығы жоқ теңқалыпты емес кодтарға **Шеннон- Фано коды** және **Хаффман коды** жатады. Бұл кодтар оқу құралының 2.3-ші және 2.4-ші бөлімдерінде сәйкесінше қарастырылады.

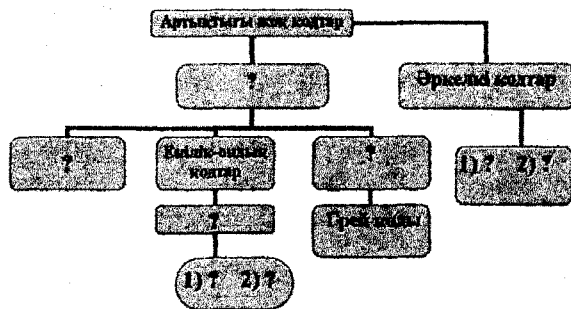
БӨЛІМ БОЙЫНША БАҚЫЛАУ СҰРАҚТАРЫ:

1. Ақпарат, хабар, байланыс арнасы, байланыс желісі дегеніміз не?
2. Хабардың қандай түрлері бар?
3. Кодтау, ақпаратты кодтау, декодтау, код, ақпарат көзін кодтау, кодтық сөз дегеніміз не?
4. Кодтау теориясының негізгі міндетіне не жатады?
5. Хабарды кодтық комбинацияға (кодтық сөзге) түрлендіру қандай мүмкіндіктер береді?
6. Қай кодтың комбинацияларында сызықша және нүкте қолданылады?
7. Трисеме коды қалай жасалады?
8. ASCII, КОИ-8 кодтарының аббревиатурасының қазақша толық аудармасы қандай?
9. Ақпарат көздерін кодтау мен арнаны кодтаудың арасында қандай айырмашылық бар?
10. Кодтың құрылымдық сипаттамаларына не жатады?
11. Код сигналдарының қандай сипаттамалары болады?
12. Теңқалыпты, тең қалыпты емес кодтарға қандай кодтаржатады?
13. Кодты сипаттайтын қандай негізгі ұғымдар бар?
14. n , m , N , N_x , K , R , R' , $W(\omega)$ анықтамаларын беріңіз.
15. Код салмағы ω анықтамасын беріңіз. Кодтық қашықтықпен (d) оның қандай байланысы бар?
16. Арнаны бастапқы ақпараттық символдарға қосымша тексеру символдарын енгізу арқылы кодтауды қалай деп атайды?
17. Импульстік белгілер саны бойынша кодтар қалай болып бөлінеді?
18. Ақпараттық символдар дегеніміз не және оның санын қалай белгілейді?
19. Қателерді түзетуге арналған кодтық комбинацияның арнайы символдар санын қалай атайды және оны қалай белгілейді?
20. Кодтарды бейнелеудің қандай тәсілдері бар?
21. Үздіксіз кодтар және блоктық кодтар қандай кодтарға жатады? Бұларда кодтау процесі қалай жүреді?
22. Бөлінбейтін блоктық кодтарға қандай кодтар жатады? Неліктен оларды бөлінбейтін деп атайды?
23. (n,k) кодтар деп қандай кодтарды айтады? n,k деп нені белгілейді?

24. Жүйеленген кодтардың барлық кодтық векторлары қалай жасалады? Жүйеленген кодтарға қандай кодтар жатады?
25. Теңқалыпты кодтарға қандай кодтар жатады? Неліктен оларды теңқалыпты деп атайды?
26. Қандай кодтар ЭЕМ-да азайту операциясын қосу операциясымен алмастыру үшін қолданылады?
27. 2-4-2-1 код пен 8-4-2-1 коды басқаша қалай аталады?
28. Бейнеленетін кодтардың басты ерекшелігі неде?
29. Грей кодының қандай ерекшеліктері мен кемшіліктері бар?
30. Артықтығы жоқ теңқалыпты емес кодтарға қандай кодтар жатады?

ӨЗДІК ЖҰМЫС ТАПСЫРМАЛАРЫ

1. Ондық санау жүйесіндегі 17, 29, 64, 100 сандарын он алтылық кодпен кодтаңыз.
2. «КОДТАУ ТЕОРИЯСЫ» хабарын Морзе кодымен кодтаңыз.
3. «COMPUTER» хабарын Трисиме кодымен, Бодо кодымен кодтаңыз.
4. Трисиме кодымен кодталған «312 111 112 213 122» хабарын декодтаңыз.
5. «ИНФОРМАТИКА» хабарын ASCII кодымен кодтаңыз.
6. ASCII кодымен кодталған « 66 79 79 75 » хабарын декодтаңыз.
7. «ҚАЗАҚСТАН» хабарын Unicode кодтауы арқылы кодтаңыз.
8. Кодтың құрылымдық және сигналдық сипаттамаларының жіктелуін схема түрінде көрсетіңіз.
9. Кодтау әдістерінің шығу тарихын көрсететін хронологиялық кесте құрыңыз.
10. Цифрлық байланыс арнасында ақпаратты жіберу процесін сипаттаңыз.
11. Ақпарат көздерінің (хабардың) 4 түріне мысалдар келтіріңіз.
12. 5 позициялы қарапайым екілік кодтың барлық комбинациялары мен оларға сәйкес келетін ондық сандарды көрсететін кесте құрыңыз. Мүмкін болатын комбинациялардың жалпы саны $N = ?$
13. 1,2,3,4,5 позициялы кез келген ондық сандарды екілік-ондық кодымен (барлық 17 нұсқасы) кодтаңыз және алынған кодты декодтаңыз.
14. 100110 екілік кодын Грей кодына, 100001 түрінде берілген Грей кодын екілік кодқа айналдырыңыз. Бұл қандай ондық сан?
15. Сұрақ белгісі бар блоктарға қандай кодтар аты жазылады?



2 АҚПАРАТТЫҢ ШЫҒУ КӨЗДЕРІН КОДТАУ

2.1 Ақпараттың шығу көздерін кодтау теоремалары

2.1.1 Шеннонның 1-ші теоремасы

Шеннонның 1-ші теоремасы ақпаратты жіберу арнасында бөгеуіл жок кезде дискретті хабарларды тиімді кодтауға болатынын айтады. Мұнда хабардың бір символына сәйкес келетін кодтық комбинациядағы екілік символдардың орташа саны хабар көзінің энтропиясына жуық болады. Оны мына мысалдағы 2.1-кестеден байқауға болады.

Кесте 2.1– Хабардың бір символына сәйкес келетін кодтық комбинация, кодтық орташа ұзындығы, хабар көзінің энтропиясы

| № | Хабар символы X_i | Ықтималдық P_{X_i} | Кодтық комбинация | Кодтық комбинация ұзындығы n_i | Кодтың орташа ұзындығы N_{ip} | Энтропия орташа мәні |
|---|---------------------|----------------------|-------------------|----------------------------------|---------------------------------|----------------------|
| 1 | X_1 | 0.14 | 10 | 2 | 0.28 | |
| 2 | X_2 | 0.16 | 001 | 3 | 0.48 | |
| 3 | X_3 | 0.16 | 011 | 3 | 0.48 | |
| 4 | X_4 | 0.15 | 101 | 3 | 0.45 | |
| 5 | X_5 | 0.12 | 110 | 3 | 0.36 | |
| 6 | X_6 | 0.11 | 111 | 3 | 0.33 | |
| 7 | X_7 | 0.09 | 1011 | 4 | 0.36 | |
| 8 | X_8 | 0.02 | 1001 | 4 | 0.08 | |
| | | $\sum P_{X_i} = 1$ | | | $\bar{n} = 2,82$ | $H = 2,85$ |

Мұның мәнісін түсіндірейік. Бастапқы хабардың A әліпбиінде (бастапқы әліпби) N символ, оның бір символға келетін орташа ақпараты $I^{(A)}$ болсын, ал екінші ретті B әліпбиі (кодтың әліпбиі) M символдан тұрсын, оның бір символға келетін орташа ақпараты $I^{(B)}$ болсын. Бастапқы әліпби символдарынан тұратын бастапқы хабарда n символ, ал кодталған хабарда m символ болсын делік. Егер бастапқы хабарда $I_{st}(A)$ ақпарат, ал кодталған хабарда $I_{mn}(B)$ ақпарат болса, онда қайта келтірілетін кодтау операциясына сәйкес кодтау кезінде ақпаратты жоғалту $I_{st}(A) \leq I_{mn}(B)$ түрінде

жинылады. Қайта келтірілетін кодтау операциясы хабардағы ақпарат санын арттырмаса, кемітпейді. Жоғарыдағы теңсіздікті символдар санының символдың орташа ақпараттық мазмұнына көбейтіндісі түрінде алмастырып жазуға болады: $n \cdot I^{(A)} \leq m \cdot I^{(B)}$ немесе $I^{(A)} \leq (m/n) \cdot I^{(B)}$

Мұндағы m/n қатынасы бастапқы әліпбидің бір символын кодтау үшін қолданылатын екінші ретті әліпбидің символдарының орташа санын сипаттайды. Мұны код ұзындығы (кодтық тізбек, кодтық вектор, кодтық сөз ұзындығы) деп атайды, $K(A,B)$ деп белгілейік. Бұдан $K(A,B) \geq I^{(A)}/I^{(B)}$

Әдетте $N > M$ және $I(A) > I(B)$ болғандықтан $K(A,B) > 1$, бұл бастапқы әліпбидің 1 символы екінші ретті әліпбидің бірнеше символымен бейнеленетіндігін білдіреді. A, B әліпбилері арқылы кодтарды құру тәсілдері көп, сондықтан солардың ішінен тиімдісін таңдау проблемасы тұрады, оны *тиімді кодтау* дейді.

Бөгеуіл болмаған жағдайдағы басты теорема деп аталатын **Шенноның 1-ші теоремасы**: Арнада бөгеуіл болмаған жағдайда бастапқы әліпбидің бір символына сәйкес келетін кодтың символдарының орташа саны бастапқы және екінші ретті әліпбилердің орташа ақпаратының қатынастарына өте жуық болатындай етіп хабарды кодтау нұсқасы әрқашан табылады.

Кодтың орташа ұзындығының мүмкін болатын минимальды мәні мынаған тең: $K^{\min}(A,B) = I^{(A)}/I^{(B)}$.

Кодтың салыстырмалы артықтығы $Q(A,B)$ мынаған тең:

$$Q(A,B) = \frac{K(A,B) - K^{\min}(A,B)}{K^{\min}(A,B)} = \frac{K(A,B)}{K^{\min}(A,B)} - 1 = \frac{K(A,B) \cdot I^{(B)}}{I^{(A)}} - 1.$$

Артықтық ұғымын қолдана отырып Шенноның 1-ші теоремасын басқаша да айтуға болады: *Бөгеуіл болмаған жағдайда әрқашан кодтың артықтығы нөлге аса жақындайтындай етіп хабарды кодтаудың нұсқасын табуға болады.*

Практикада екілік кодты қолдану ($M=2$) ыңғайлы, бірдей ұзақтық пен ықтималдық болғанда әрбір элементар сигналда (0 немесе 1) 1 бит ақпарат болады ($\log_2 M=1$); бұдан келесі теңдік шығады: $K^{\min}(A,2) = I^{(A)}$.

Осының негізінде Шенноның 1-ші теоремасын келесі түрде түсіндіруге де болады: *Бөгеуіл болмаған жағдайда екілік кодтың орташа ұзындығы бастапқы әліпбидің символына сәйкес келетін орташа ақпаратқа аса жуық болады.*

2.1.2 Шенноның 2-ші теоремасы

Шенноның 2-ші теоремасы арнада бөгеуіл болған жағдайда хабарды берілген шынайылықпен қабылдаушыға жеткізуге мүмкіндік беретін кодтау жүйесін табуға болатынын айтады. Ол үшін арнаның өткізу қабілеті хабарды жасау жылдамдығынан үлкен немесе оған тең болуы керек.

Келесі белгілеулерді қолданайық.

K – хабар көзінен жасалып шыққан блок ұзындығы.

L – кодталған соң арнамен жіберілетін блок ұзындығы.

R – хабарды жіберу жылдамдығы (хабар көзінің өнімділігі).
 $R=K/L$.

C – арнаның өткізу қабілеті, (басқаша айтқанда арнаның кірісі мен шығысындағы өзара ақпараттың максимумы), $C=\max(I(X,Y))$, мұндағы X пен Y — кездейсоқ шама түріндегі арнаның кірісі мен шығысы.

P_{er} – блокты декодтау қатесінің орташа ықтималдығы.

$P_{er,max}$ – блокты декодтау қатесінің максимальды ықтималдығы.

$P_{er,max} = \max P_{er}$.

Тура теорема: Егер хабарды жіберу жылдамдығы арнаның өткізу қабілетінен аз болса ($R < C$), онда жіберілетін блок ұзындығы шексіздікке ұмтылғанда декодтау қатесінің орташа және максимальды ықтималдықтары нөлге ұмтылатындай мынандай кодтар $\{x_1, x_2, \dots, x_M\}$ және декодтау әдістері табылады, яғни $L \rightarrow \infty$ болғанда, $P_{er} \rightarrow 0$, $P_{er,max} \rightarrow 0$

Кері теорема: Егер хабарды жіберу жылдамдығы арнаның өткізу қабілетінен үлкен болса ($R > C$), онда жіберілетін блок ұзындығы артқанда ($L \rightarrow \infty$) декодтау қатесінің ықтималдығы нөлге ұмтылатындай $P_{er} \rightarrow 0$ жіберу тәсілі табылмайды.

2.2 Тиімді кодтар

Бір хабарды әртүрлі тәсілдермен кодтауға болады. Хабарды жіберуге өте аз уақыт жұмсайтын кодты **тиімді код** деп санайды.

Егер әрбір элементар символды (0 және 1-ді) жіберуге бірдей уақыт жұмсалатын болса, онда ұзындығы мүмкіндігінше өте аз код **тиімді код** болып табылады. Бір әріпті таңбалау үшін орташа есеппен ең аз элементар символ қолданатын кодтар бар, оларға Шеннон-Фано мен Хаффманның кодтары жатады.

Тиімді кодтарды құрастырудың принципі:

1. Әрбір элементар символ ең көп ақпаратты тасымалдауы тиіс, ол үшін: кодталған мәтіндегі олардың кездесу жиілігі орташа есеппен бірдей болуы керек. Сонда ең үлкен энтропия болады.
2. Бастапқы әліпбидің үлкен ықтималдықтары бар (немесе хабарда кездесу жиілігі жоғары) әріптеріне (яғни бастапқы ақпаратқа) екінші реттегі әліпбидің (яғни кодтың) ең қысқа кодтық сөздерін (комбинацияларын) сәйкес келтіру керек.

2.3 Шеннон-Фано коды

Шеннон-Фано әдісі бойынша кодтаудың негізіне салынған басты принцип кодтық комбинацияларды жасау кезінде ондағы ақпараттың көлемін неғұрлым көп етуге ұмтылу керектігінде, яғни барлық алдындағы сандардың мәніне қарамастан, бұл сан бірдей мүмкіндікпен ол үшін болуы ықтимал мәндердің (0 немесе 1) екеуін де қабылдауында. Кодтық комбинациялардың ұзындықтары әркелкі, яғни бұл код теңқалыпты болмайды. Шеннон-Фано әдісі бойынша хабардағы кездесу жиілігі жоғары, яғни үлкен ықтималдықтары бар әріптерге қысқа кодтық комбинациялар сәйкес келтіріледі, ал сирек кездесетін әріптер ұзын кодтық комбинациялармен кодталады.

Хабар әріптерін Шеннон-Фано кодымен кодтау үшін кодтық комбинациялар мына ретпен құрастырылады:

1. Хабар әріптері мен олардың ықтималдықтары кестеге ықтималдықтарының кему реті бойынша жазылады.
2. Кесте екі бөлікке бөлінеді, олардың екі бөліктегі ықтималдықтарының қосындысы бір-біріне жақын болуы тиіс.

3. Әр бөлік 0 және 1 символдары арқылы белгіленеді. Кестенің жоғарғы бөлігіне кодтық сөздің ең үлкен биті ретінде 0, ал төменгісіне 1 жазылады.
4. Ары қарай олар тағы да ықтималдықтары тең бөліктерге бөлініп, әр бөлік 0 және 1 символдары арқылы белгіленіп отырады. Кестені бөлу соңғы кодтық комбинация алынғанға дейін қайталана береді.

Мысал. Шеннон-Фано кодымен кодтау.

| Хабар әріптері | Ықтималдықтары | 1-ші қадам | 2-ші қадам | 3-ші қадам | 4-ші қадам | Шеннон-Фано кодының кодтық комбинациялары | |
|----------------|----------------|------------|------------|------------|------------|---|------|
| X_i | P_i | | | | | | |
| X_1 | 1/4 | 0 | 0 | ----- | ----- | 00 | |
| X_2 | 1/4 | | 1 | ----- | ----- | 01 | |
| X_3 | 1/8 | 1 | 0 | 0 | ----- | 100 | |
| X_4 | 1/8 | | | 1 | ----- | 101 | |
| X_5 | 1/16 | | 0 | 1 | 0 | ----- | 1100 |
| X_6 | 1/16 | | | | 1 | ----- | 1101 |
| X_7 | 1/16 | 1 | 0 | 1 | 0 | 1110 | |
| X_8 | 1/16 | | | | 1 | 1111 | |

Жасалынған кодтың орташа ұзындығы энтропияға тең болып шықты. Кодтың бұлайша сәтті жасалуы ықтималдықтардың теңықтималды топтарға тура бөлінгендігінде. Алынған кодтың тиімділігі энтропияның кодтың орташа ұзындығына қатынасы ретінде анықталады. Тиімділік 1-ге тең болған жағдайда код тиімді болады. Егер бұл хабарды біркелкі ұзындықтағы кодпен кодтасақ, тиімділігі анағұрлым аз болар еді.

Шеннон-Фано коды префикстік болып табылады, яғни ешбір комбинация өзінен ұзын комбинацияның басы болмайды.

Префикстік кодтар. Ұзындықтары әр түрлі кодтық комбинацияларды жібергенде олардың арасына ажырату белгілерін қою қажет болады. Кодтық комбинациялардың арасына бір-бірінен ажырататын белгі оймай хабарды кодтағанда, оларды бастапқы жіберілген хабарды алатындай етіп бірмәнді түрде декодтау проблемасы тұрады. Осы проблеманы шешуге арналған

ең қарапайым, әрі жиі қолданылатын кодтарға *префикстік кодтар* жатады.

Префикстік кодтарға мына шарт (*Фано шарты*) орындалуы тиіс: *теңқалыпты емес кодты бірімәнді түрде декодтауға болады, ол үшін ешбір кодтық комбинация өзінен ұзын басқа комбинациялардың басымен (префикс) бірдей болмауы тиіс.*

Префикстік код мысалы: 00, 01, 100, 101, 1100, 1101, 1110, 1111

Префикстік емес код мысалы: 0, 00, 01, 10, 11, 000, 100, 101, 110, 111

2.4 Хаффман коды

Хаффман кодымен ақпаратты тиімділеу. Хаффман коды ұзындығы қалыпты емес (кодтық сөздегі разрядтар саны әркелкі) кодтар тобына жатады. Бұл алгоритмді Дэвид Хаффман 1952 жылы ұсынған болатын.

Хаффман коды да префикстік болып табылады, яғни ешбір комбинация өзінен ұзын комбинацияның басы болмайды, ал бұл бір-бірінен ажырататын белгілерсіз символдарды бірімәнді декодтауға мүмкіндік береді.

Хаффман коды ақпараттық артықтықты азайтып, жад көлемін кішірейтетін және ақпараттың байланыс арнасы арқылы жіберілу жылдамдығын арттыратын тиімді код болғандықтан, үлкен қолданысқа ие болды. Ақпаратты сығу программаларында (pkzip2, LZH т.б.) қолданылады.

Хаффман коды хабардағы жиі кездесетін символдарды қысқа кодтармен, ал сирек кездесетін символдарды ұзын кодтармен алмастыру арқылы ақпарат көлемін 80%-ға дейін сығуға мүмкіндік береді.

Хаффман кодының тиімділігі Шеннон-Фано кодынан кем емес. Хаффман коды әріптердің хабардағы кездесу ықтималдықтарын, өзара тең емес болатын сигналдардың статистикалық қасиеттерін ескереді.

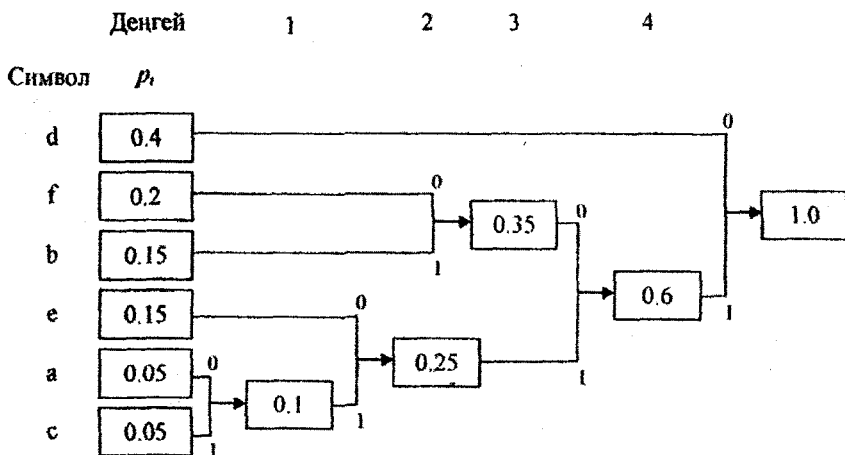
Хаффман кодының жасалу жолы. Хаффман кодымен кодтау үш кезеңде жүзеге асады.

Бірінші кезеңде мына 1.4-суреттегідей ақпарат көзінің барлық кодталатын әріптері ықтималдықтарының кемуі ретімен бір бағанға жазылады. Бағандағы ықтималдықтары ең кіші соңғы 2 әріптің

Ықтималдықтарын қосып, бағанды жана ықтималдықтардың кемуі бойынша қайтадан жазады, содан кейін бірінші қосалқы элементтер тобы құрылады.

Әрі қарай осылай ең кіші ықтималдығы бар элементтер жұптарын қосып, келесі қосымша топтарды құру процесі қосындының нәтижесінде ықтималдылығы 1-ге тең элементті алғанға дейін қайталады.

Жолдың басындағы әріп 1, ал соңғысы 0 деп кодталады.



2.1-сурет. Хаффман кодтауы, $n=4$

Жұмыстың бірінші кезеңінің нәтижесінде қосымша топтардан мынандай 2.1-кесте жасалынады.

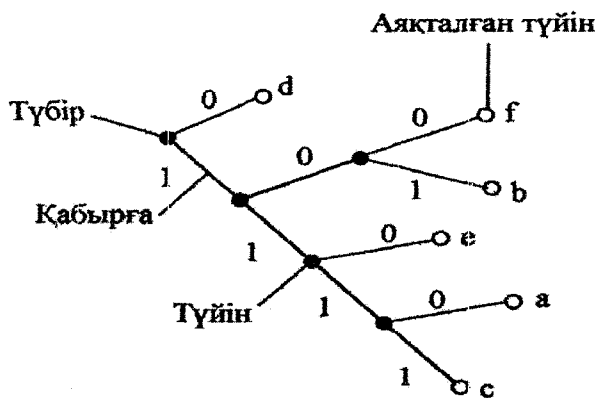
Кесте 2.1 – Ықтималдықтар және екі символдың энтропиясы

| X_i | a | b | c | d | e | f |
|-----------------|-----------------------------|----------|----------|----------|----------|----------|
| P_i | 0.05 | 0.15 | 0.05 | 0.4 | 0.2 | 0.15 |
| $I(p_i)$ бит | 4.32 бит | 2.74 бит | 4.32 бит | 1.32 бит | 2.32 бит | 2.74 бит |
| $H(X)$ бит | Энтропия ≈ 2.25 бит | | | | | |

Екінші кезеңде қосымша топтар кестесі бойынша берілген хабар үшін кодтық комбинациялар жасалынады. Кодтық комбинацияны жасау үшін кодтық ағаш құрылады. Кодтық ағашта жоғарыдан төмен қарай бағытта жылжи отырып, әрбір әріптің өзіне сәйкес келетін кодтық комбинациясын жазуға болады. Кодтық ағаштың төбесін белгілеп, оған 1.0 ықтималдығын жазады. Бұтақтың төбесінен түйіндермен аяқталатын екі бұтақты жүргізеді. Түйіндерге қосындысы бірді беретін ықтималдықты жазады. Тік бұтаққа (ең аз ықтималдықты) «0» (ноль) символын, ал көлденең кететін бұтаққа (ең көп ықтималдықты) – «1» символын жазады. Түйіндерге жиынтықтау кезінде бірді беретін ықтималдықты жазады.

Егер кейбір түйінде ықтималдық қандай да бір хабардың ықтималдығына тең келсе, бұл аяқталған түйінге осы хабардың символын жазады да, келесі түйінді талдауға көшеді. Егер түйіндегі ықтималдық екі ықтималдықтың қосындысынан құрастырылса, онда одан тағы да түйіндермен аяқталатын екі бағытты бұтақты салады. Түйіндерге қабылданған бағытты ескере отырып, бұтақтарға тігінен – «0» (ноль), көлденеңінен – «1» символдарын келтіріп жазады.

Ақпарат көзінің барлық хабарының символдары ағаш төбесіндегі түбір түйінге келтірілгенше 2.2-суретте көрсетілгендей кодтық ағашты құру процесі жалғасады.



2.2-сурет. Кодтық ағашты құру процесі

Үшінші кезеңде сәйкес хабардың кодтық комбинациясы келесі түрде жазылады. Жасалынған кодтық ағашта бұтақтар бойынша төбедегі түйіннен бастап, сәйкес хабардың әрпіне сәйкес келетін түйініне дейін солдан оңға қарай 1 немесе 0 символын кезекпен жазамыз. Алынған кодтық комбинация Хаффман кодының тиімді жолын білдіреді 2.2-кестеде кодтық ағашты қолданып кодтық комбинацияны жазу көрсетілген.

Кесте 2.2 – Кодтық ағашты қолданып кодтық комбинацияны жазу

| Хабар символы X_i | Ықтималдық P_i | Кодтық сөз | Кодтық сөздің ұзындығы | Кодтық сөздің орташа ұзындығы |
|------------------------|---------------------|------------|------------------------|--|
| d | 0.4 | 0 | 1 | $\bar{n} = \sum_i P_i n_i$ $\bar{n} = 2.3 \text{ бит}$ |
| f | 0.2 | 100 | 3 | |
| b | 0.15 | 101 | 3 | |
| e | 0.15 | 110 | 3 | |
| a | 0.05 | 1110 | 4 | |
| c | 0.05 | 1111 | 4 | |

Қарастырылған мысалдағы кодтық сөздің орташа ұзындығы $\bar{n} = 2.3$ бит, яғни энтропияға жақын, энтропия $H(X) = 2.25$ битке тең.

Бұл код кодтық сөздің минималды орташа ұзындығына ие болады. Кодтық сөздің орташа ұзындығы сәйкес ықтималдықпен анықталады.

Код тиімділігі немесе *сығу факторы* деп аталатын маңызды шама бар, ол кодтың орташа ұзындығының энтропияға қатынасы арқылы табылады:

$$\eta = \frac{H(x)}{\bar{n}}$$

Жоғарыда қарастырылған мысалдағы кодтың тиімділігі η (ню) = 0.976 болды.

Мысалдан көрініп тұрғандай, символдар ықтималдықтарының арасындағы айырмашылық жоғары болған сайын, қарапайым блоктық кодтармен салыстырғанда Хаффман кодының тиімділік жағынан ұтысы жоғары болады.

Хаффман кодының декодталуы. Хаффман кодының декодерінің кодтық ағашы 1.5-суретте көрсетілген. Әрбір жаңа сөздің декодталуы кодтық ағаштың бастапқы түйінінен (яғни, түбірден) басталады. Егер біз «0» қабылдасақ, онда нөлге сәйкес келетін қабырғамен жүреміз (жоғарғы қабырға бойынша). Біздің мысалда бұл жағдайда «d» аяқталған түйінге жетеміз. Демек, d символы жіберілді және біз жаңа символдың декодталуын түбірден бастаймыз.

Егер біз «1» қабылдасақ, онда төменгі қабырға бойынша жүреміз және екі қабырға шығатын түйінге келіп жетеміз. Келесі қабылданған бит осы екі қабырғаның қайсысын таңдау керек екенін көрсетеді. Бұл процедура аяқталған түйінге жеткенше қайталанатын. Осы жағдайда жіберілген символ туралы шешім қабылдап, қайтадан кодтық ағаштың түбіріне өтеміз.

БӨЛІМ БОЙЫНША БАҚЫЛАУ СҰРАҚТАРЫ:

1. Егер әрбір элементар символды (0 және 1-ді) жіберуге бірдей уақыт жұмсалатын болса, онда ұзындығы мүмкіндігінше өте аз және хабарды жіберуге өте аз уақыт жұмсайтын код қалай аталады?
2. Шенноның 1-теоремасы не туралы айтады?
3. Шенноның 1-теоремасы бойынша хабардың бір символына сәйкес келетін кодтық комбинациядағы екілік символдардың орташа саны қандай болады?
4. Артықтық ұғымын қолдана отырып Шенноның 1-ші теоремасын басқаша қалай айтуға болады?
5. Шенноның 2-теоремасы бойынша арнаның өткізу қабілеті қандай болуы керек?
6. Шенноның 2-ші тура және кері теоремалары қалай айтылады?
7. Тиімді код дегеніміз не?
8. Шеннон-Фано мен Хаффманның кодтары қандай кодтарға жатады?
9. Қай кодты құру үшін хабар әріптері мен олардың ықтималдықтары кему реті бойынша жазылған кесте екі бөлікке бөлінеді?
10. Қай код ақпарат көлемін 80%–ға дейін сығуға мүмкіндік береді? Ақпаратты сығудың қандай программаларында қолданылады?
11. Қай кодта кодтық комбинацияны жасау үшін кодтық ағаш құрылады?
12. Код тиімділігі немесе сығу факторы деген не? Ол шама қалай анықталады?
13. Хаффман кодының декодталуы неден басталады?
14. Хаффман кодының кодтық ағашын қалай құрады?
15. Префикстік код дегеніміз қандай код?

ӨЗДІК ЖҰМЫС ТАПСЫРМАЛАРЫ

1. Бастапқы әліпби екі әріптен (А,Б) тұрады, ықтималдықтары $P(A)=0.8$, $P(B)=0.2$. Осы әріптердің 2 әріптік комбинацияларынан тұратын хабарды тиімді кодтау үшін Шеннон-Фано кодының комбинацияларын жазыңыз.

| Хабар әріптері | Ықтималдық | Топтарға бөлу | | | Шеннон-Фано коды |
|----------------|------------|---------------|---|---|------------------|
| АА | 0,64 | 0 | - | - | |
| АБ | 0,16 | 1 | 0 | - | |
| БА | 0,16 | | 1 | 0 | |
| ББ | 0,04 | | | 1 | |

2. Бастапқы әліпби әріптері мен олардың хабарда кездесу жиіліктері берілген. Шеннон-Фано кодын құрыңыз.

| x_1 | x_2 | x_3 | x_4 | x_5 | x_6 | x_7 | x_8 | x_9 |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 0.21 | 0.15 | 0.15 | 0.14 | 0.13 | 0.08 | 0.06 | 0.06 | 0.02 |

3. Бастапқы әліпби әріптері мен олардың хабарда кездесу жиіліктері берілген. Хаффман кодын, кодтық ағашын құрыңыз.

| x_1 | x_2 | x_3 | x_4 | x_5 | x_6 | x_7 | x_8 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| 1/4 | 1/4 | 1/8 | 1/8 | 1/16 | 1/16 | 1/16 | 1/16 |

4. Бөлімдегі 2.1-суреттегі кодтық ағашты пайдаланып dddcfbddd хабарын Хаффман кодымен кодтаңыз. Кодтық сөздің орташа ұзындығын, кодтың тиімділігін есептеңіз.

5. Бөлімдегі 2.1-суреттегі кодтық ағашты пайдаланып Хаффман кодымен кодталған 1001110111110 хабарын декодтаңыз.

6. Префикстік код комбинацияларына мысалдар келтіріңіз.

7. Қайсы код (Код1,Код2,Код3,Код4) префикстік болады?

| Символ | Код1 | Код2 | Код3 | Код4 |
|--------|------|------|------|------|
| x_1 | 0 | 0 | 0 | 0 |
| x_2 | 0 | 00 | 01 | 10 |
| x_3 | 1 | 1 | 011 | 110 |
| x_4 | 11 | 11 | 0111 | 111 |

8. Бастапқы әліпби үшін ықтималдықтар белгілі: $p(a_1) = 0,13$; $p(a_2) = 0,16$; $p(a_3) = 0,02$; $p(a_4) = 0,03$; $p(a_5) = 0,6$; $p(a_6) = 0,01$; $p(a_7) = 0,05$. Тиімді кодтың кодтық комбинациясының орташа ұзындығы неге тең?

3 ДЕРЕКТЕРДІ СЫҒУ

3.1 Деректерді сығу ұғымы

Байланыс арнасы арқылы жіберілетін және сақталатын деректердің көп бөлігінің көлемі үлкен болғандықтан, ақпаратты жіберу, сақтау және өңдеу үлкен шығындарды қажет етеді. Бұл деректер негізінен оның қарапайым қолданылуын қамтамасыз ететін түрде сақталады (мысалы: қарапайым мәтіндер, мәтіндік редактордың ASCII коды, ЭЕМ деректерінің екілік коды, деректерді жинау жүйесіндегі сигналдардың жеке саналымы және т.б.) Алайда, деректерді бейнелеудің мұндай қарапайым қолданысы деректерді сақтау мен оларды жіберу үшін бірнеше есе көп орынды қажет етеді. Сондықтан деректерді сығу – бұл радиотехниканың қазіргі заманғы өзекті бағыттардың бірі.

Деректерді сығу. Ақпараттың мазмұнын өзгертпей ақпарат көлемін азайту процесін ақпаратты сығу немесе деректерді сығу немесе **компрессия** немесе **архивация** деп атайды. Сығу, мысалы, дискке көп ақпарат жазуға, қатты диск көлемін ұлғайтуға, модеммен жұмыс жасауды тездетуге және т.б. мүмкіндік береді.

Деректерді сығудың екі негізгі типі бар: **жоғалтусыз сығу** (немесе **шығынсыз, толығымен қайтарымды сығу**) және **2) жоғалтумен сығу**. Жоғалтумен сығуда ақпараттың маңызы аз бөлігі алынып тасталынады және ақпарат толығымен бастапқы калпына келтірілмейді.

Бірінші типті әдетте деректерді қайтадан алпына келтіру маңызды болған жағдайда, көбінесе мәтіндер және цифрлы мәліметтер үшін қолданылады.

Ал, екінші тип негізінен бейнелер мен дыбыстарға қолданылады. Маңызы төмен бөліктерін алып тастау арқылы өте тиімді, жоғары дәрежелі сығу жүреді.

Деректерді сығу жүйесінің шартты құрылымы келесі түрде жүреді: **Ақпарат көзінің деректері → Кодер → Сығылған деректер → Декодер → Қалпына келтірілген деректер.**

Деректерді сығудың мақсаты – ақпарат көзі шығаратын деректерді байланыс арналары арқылы тиімді жіберуді және сақтауды қамтамасыз ету, яғни **ақпарат көздерін оңтайлы кодтау**

арқылы ақпаратты сақтауға немесе тасымалдауға көмекші техникалық шығындарды минимализациялау болып табылады.

Хабарда артықтық болған жағдайда ақпаратты сығуға болады.

Артықтық табиғи және жасанды артықтық болып екі түрге бөлінеді, ал табиғи артықтықтың өзі семантикалық және статистикалық болып бөлінеді. Табиғи артықтық бастапқы әліпбимен, ал жасанды артықтық екінші ретті әліпбимен байланысты болады.

Ақпаратты жоғалтпайтындай етіп неғұрлым қысқа айтылған семантикалық артықтық болады. Ал статистикалық артықтық теңқымталды емес болып үлестірілген бастапқы әліпбидің сапалық белгілері мен олардың өзара байланыстылығына тәуелді болады.

Ақпаратты сығу үшін арнаулы сығушы программалар (архиваторлар) қолданылады. Кеңінен тараған архиваторлар RAR, WINRAR, WINZIP және т.б. жатады. Төмендегі 3.1-кестеде сығылған файлдардың кеңейтілімдері, соларға сәйкес программалар архиваторлар және сығу әдістері көрсетілген.

Кесте 3.1– Сығылған файлдардың кеңейтілімдері, соларға сәйкес программа-архиваторлар және сығу әдістері

| Файлдардың кеңейтілімдері | Сығушы программалар (архиваторлар) | Қолданылатын деректерді сығу (кодтау) әдістері |
|---------------------------|------------------------------------|--|
| z | compress | LZW |
| arc | arc, pkarc | LZW, Хаффман |
| zip | zip, unzip, pkzip, pkunzip | LZW, LZ77, Хаффман коды, Шеннон-Фано коды |
| gz | gzip | LZ77, Хаффман коды |
| bz2 | bzip2 | Берроуз-Уиллер, Хаффман коды |
| arj | arj | LZ77, Хаффман коды |
| ice, lzh | lha, lharc | LZSS, Хаффман коды |
| pak | pak | LZW |

Деректерді сығу ақпаратты сақтағанда, резервтік көшіру жасағанда, радиобайланыста, телекоммуникациялық желі арқылы ақпаратты тасымалдағанда, цифрлық теледидарда қолданылады. Сығылған деректер желіде жылдам жіберіледі. Деректерді сығу радиобайланыста, цифрлық теледидарда аудиосигналдар мен бейнесигналдарды кодтау арқылы іске асады.

Графигтік ақпараттарды сақтауға арналған файл форматтарының бірлығында деректерді сығу қолданылады. 12-кестіде графигтік файлдардың кеңейтілімдері мен сығу әдістері көрсетілген.

3.1-кесті - графигтік файлдардың кеңейтілімдері мен қолданылатын сығу әдістері

| Файлдың кеңейтілімі | Қолданылатын деректерді сығу (кодтау) әдістері |
|---------------------|--|
| LZW | LZW, Хаффман коды |
| | Шығыны бар кодтау, Хаффман коды, арифметикалық кодтау |
| | RLE (<i>Run Length Encoding</i> , айнымалы ұзындықты кодтау) – қарапайым сығу әдісі, келесі байтты неше рет қайталау керектігін көрсететін арнайы код-белгіні айрықшалауға негізделген. |
| | факстар үшін CCITT/3, LZW, т.б. |

Ақпаратты сығу дәрежесі сығусыз және сығу әдісін қолданып ақпаратты сығу немесе тасымалдауға кеткен шығындармен байланысты. Сығу дәрежесі қолданылатын алгоритмге және ақпараттың өлшеміне байланысты болады.

Мысалы, практикада ZIP сығу бағдарламасы көмегімен WORD редакторының форматындағы мәтіндік деректерді сығу және 10 пайызға дейін жетеді.

Ақпаратты сығу дәрежесі келесі 3.1-формуламен анықталады:

$$G_k = \frac{k_0 - k_m}{k_0} \quad (3.1)$$

Мұнда k_0 – сығусыз, ал k_m – сығу әдісін қолданып ақпаратты тасымалдауға кеткен шығын.

Шығыны бір кодтау және шығынсыз кодтау. Ақпаратты сығу әдісін ұғым қарастырылады, соған сәйкес шығыны бар кодтау және шығынсыз кодтау қолданылады.

1-ші ұғым маңызды емес ақпарат – бұл тасымалдағанда ақпараттың кетуге болатын ақпарат түрі. Мысал ретінде

дәстүрлі телефонияны келтіруге болады. Телефон арналарында ақпарат 3,4 кГц жолағында тасымалданады. Қалған спектральды компоненттері, яғни дыбыстың естілмейтін бөлігі шеттеліп қалады, бұл арада тасымалданатын ақпараттың бір бөлігі жоғалып кетеді. Бастапқы дыбыстық сигнал қабылдайтын жақта толығымен қалпына келе алмайтыны айқын. Бұл жағдай шығыны бар кодтау болады.

2-ші ұғым – артықтық ұғымы ақпарат қабылдаушыға қажетті хабардың бірнеше рет қайталануын білдіреді. Мұндай артықтықты ақпаратты жоғалтпай-ақ келтіруге болады. Мысал ретінде Хаффман кодтауын айтуға болады. Мұндай кодтау түрін шығынсыз кодтау деп атайды.

3.2 Сығу түрлері, сығу алгоритмдерінің жіктелуі

3.2.1 Сығу түрлері

Деректерді сығуда тек қана мәтін емес сонымен қатар бейне, аудио және бейнелерді сығуды жүзеге асыруға болады. Олардың әрқайсысының өзінің сығу әдістері бар. Соның ішінде мәтінді сығуды қарастырайық. Мәтінді сығудың үш түрі бар: *қысқарту*, *қарапайымдылау*, *жалпылау*.

Қысқарту бұл – қайталауларды алып тастау, синоним сөздерді қысқарту, сөйлемнің фрагменттерін қысқарту, бір немесе бірнеше сөйлемдерді қысқарту (сөйлемдерді қысқартқан кезде тек бағыныңқы сөйлемдерді қысқартуға болады).

Қарапайымдылау – сөз тіркестерін сөзбен ауыстыру, бірнеше сөйлемдерді бір сөйлемге біріктіру, күрделі бағыныңқы сөйлемдерді қарапайым сөйлемдерге ауыстыру.

Жалпылау – бірыңғай сөйлем мүшелерін жалпы бір атаумен өзгерту, сөйлемнің бір мүшесін немесе сөйлемнің өзін жалпыландырылған бір атаумен ауыстыру.

Деректерді сығу түрлерін сипаттайық.

Статикалық деректерді сығу (static data compression) – ұзақ сақтау мен архивация үшін қолданылады, арнайы сервистік программа-архиваторлар (мысалы, ARJ, PKZIP/PKUNZIP)

көмегімен орындалады. Қайта қалпына келтіруден (декомпрессия, декодтау) кейін сығылған ақпарат бастапқы қалпына келеді.

Динамикалық сығу (нақты уақыттағы сығу; dynamic compression, compression in real time) – ЭЕМ-ның сыртқы құрылғыларына (соның ішінде монитор экранына да) шығарылатын және жедел қолжетімділікті талап ететін деректердің дисктегі жад аймағында алатын орнын қысқартуға арналған. Деректерді динамикалық сығу мен оларды қайта қалпына келтіру арнайы программалық құралдармен автоматты түрде және «лезде» жүргізіледі.

Бейнені сығу (image compression) – бұл динамикалық сығудың бір түрі, графикалық бейнелерді (сурет, схема, сызба) тасымалдауыштарда сақтағанда көлемін қысқартудың әдісі немесе техникалық тәсілі, сығуда бейне элементтерін кодтаудың әртүрлі әдістері қолданылады. Байланыс арнасымен факс жібергенде, мультимедиа жүйелерінде, видеофондарда қолданылады.

Дискті сығу (disk compression) – арнайы қолданбалы программалар көмегімен (DoubleSpace, Stacker, SuperStor сияқты) дискке жазу кезінде динамикалық сығуға негізделген техникалық тәсіл, ал дисктен оқыған кезде автоматты түрде бастапқы қалыпқа келтіреді.

Физикалық сығу (physical compression) – ішіндегі ақпараттың сипатын есепке алмай деректерді «формальды» түрде жинақы түрге келтіретін сығу әдіснамасы.

Физикалық сығу әдістері 2 топқа бөлінеді:

1) *бүкіл файлды сығу* – гаг, zip сияқты және т.б. сығу программасы арқылы файлдың бүкіл деректері оқылып, сығылады, нәтижесінде жаңа файл құрылады, сығу ұтысы мардымсыз болғанымен, мұндай сығылған файл ұзақ мерзімге сақтау немесе жіберу үшін қолданылады.

2) *файл құрамына енгізілген сығу* – мұндай форматтағы файлдарды оқуға арналған программалар файлдарды ашу кезінде деректерді оқиды, файлдардың ішкі сығылуын графикалық файлдарда (әдетте растрлық суреттер көп орын алады), файлда үлкен көлемді қайталанатын деректер болғанда қолдану тиімді.

Логикалық сығу (logical compression) – алфавиттік, цифрлық немесе екілік символдардың жинағы сәйкес басқа жинақпен алмастырылатын әдіснама. Бұл әдісте бастапқы деректердің

мағыналық мәні сақталады. Әдістің мысалына сөз тіркестерін аббревиатурамен (бастапқы әріптерін ғана қалдырып қысқартылған сөздер) алмастыру жатады. Логикалық сығу символдық немесе жоғары деңгейде жүргізіледі және тек қана бастапқы деректердің мазмұнына негізделген, бейнелерге (суреттер) қолданылмайды.

Симметриялық сығу (symmetric compression) – деректерді сығу мен қайта бастапқы қалпына келтіру алгоритмдерінің жасалу жолдары жақын немесе өзара тығыз байланысқан түрде болатын сығу әдіснамасы. Симметриялық сығуды қолдану кезінде деректерді сығу мен қайта бастапқы қалпына келтіруге жұмсалатын уақыт шамалас болады. Деректермен алмасу бағдарламаларында әдетте осы симметриялық сығу қолданылады.

Асимметриялық сығу (asymmetric compression) – «бір бағытта» жұмыстарды жасауға (мысалы, компрессияға) жұмсалатын уақыт баска бағытта жұмыс жасауға (мысалы, декомпрессияға) жұмсалатын уақыттан артық болатын әдіснама. Мысалы, бейнелерді (сурет) сығу үшін әдетте сығылған ақпаратты ашуға қарағанда уақыт пен жүйелік ресурстар көп жұмсалады. Бұл тәсілдің тиімділігі мынада: бейнені сығу тек бір рет қана орындалады, ал сығылған бейнеге декомпрессияны көп мәрте орындауға болады. Ал «кері бағыттағы» асимметриялық алгоритмдер (декомпрессияға қарағанда деректерді сығуға аз уақыт жұмсайтын) деректердің резервтік көшірмелерін жасағанда қолданылады.

Адаптивті кодтау (adaptive encoding) – деректерді сығу үшін қолданылатын деректердің нақты бір түріне алдын ала бейімделмейтін кодтау әдіснамасы. Адаптивті кодтауды қолданатын бағдарламалар деректер көлемін максималды қысқарта отырып деректерді сығудың кез келген түріне бейімделе алады.

Адаптивті емес кодтау (nonadaptive encoding) – деректердің нақты бір типін немесе типтерін ғана сығуға бағытталған кодтау әдіснамасы. Осы принцип бойынша құрылған кодтауыштардың құрамында кодталатын деректерде жиі кездесетін «алдын ала анықталған жолдардан» тұратын статикалық сөздіктер болады. Бұл кодтаудың мысалына Хаффманның сығу әдісі жатады.

Жартылай адаптивті кодтау (half-adaptive coding) – деректерді сығу кезінде адаптивті және адаптивті емес кодтау элементтерін пайдаланатын кодтау әдіснамасы. Бұл кодтауда

кодтаушы операциялардың екі тобын орындайды: алдымен — кодталатын деректер массивін қарап шығады және оларға сөздік құрады, содан кейін — кодтаудың өзі орындалады.

Жоғалтусыз немесе шығынсыз сығу (lossless compression) — алдын ала кодталған деректер қайта ашудан кейін ешбір өзгеріссіз толық қайта қалпына келтірілетін сығу әдіснамасы.

Жоғалтумен немесе шығынмен сығу (lossy compression) — бастапқы деректерді сығудың максималды дәрежесін қамтамасыз ету үшін оның құрамындағы деректердің бір бөлігі лақтырылып тасталынады. Мұндай сығу әдісін жүзеге асыратын бағдарламаларды мәтіндік, сандық және кестелік деректер үшін қолдануға болмайды. Алайда, графикамен жұмыс жасайтын программамалар үшін бұл орынды болып табылады. Қайта қалпына келтірілген суреттің сапасы графикалық материалдың сипаты мен бағдарламада жүзеге асқан сығу алгоритмінің нақтылығына байланысты болады. Қайта қалпына келтірілген суреттің сапасы графикалық материалдың сипаты мен бағдарламада жүзеге асқан сығу алгоритмінің дұрыстығына байланысты болады. Мұндай сығу алгоритмдерінің кейбір түрлерінде қайта қалпына келтірілген суреттерді пайдаланған кезде (мысалы оны монитор экранында қарау кезінде, полиграфияда, принтермен баспаға шығаруда) бастапқы графикалық бейненің мүмкін болатын жоғалту деңгейі ескеріледі. Сондықтан да бұл әдістерді «шығындарды минимизациялап сығу» деп те атайды.

Ақпаратты сығуды қолданудың қажеттілігіне байланысты сығу тәсілдері кең таралған. Алайда екі маңызды мәселе бар. Біріншіден, кең таралған сығу тәсілдері көнерген және сығудың қажет дәрежесін қамтамасыз етпейді. Сонымен қатар, олар көптеген бағдарламалық өнімдер мен кітапханаларға енгізілген және сондықтан да ұзақ қолданылады. Екінші мәселе деректердің сипаттамаларына сәйкес келмейтін сығу тәсілдерін жиі қолдану болып табылады.

Ақпарат түрлеріне байланысты сығудың түрлерін қарастырайық.

1. Әріпмен сығу:

- тиімді кодтар (Шеннон-Фано, Хаффман кодтары);
- блоктық кодтар;
- негізі екіден артық кодтауға өту.

2. **Сөздер мен сөз тіркестерін сызу:**
- аббревиатура (АҚШ, АСЦИ, ЭЕМ);
 - иероглифтер;
 - сөздің жалғауларын алып тастау;
 - жиі қайталанатын әріптерді алып тастау;
 - әріптерді таңдамалы түрде алып тастау;
 - лексикалық кодтау (жеке лексемалар екілік кодпен алмастырылады).

3. **Мәтінді сызу және жималау:**
- Библиографиялық сипаттама (авторы, атауы...);
 - Аннотация (2/3 бетке дейін);
 - Реферат (16 параққа дейін).

4. **Сандар массивін сызу.** Мұның мысалына Бабко әдісі жатады. Мекемелерде номенклатура ондық цифрлармен нөмірленеді (кодталады). Сандар массивін өсу ретімен орналастырса, әрбір келесі санның алдыңғысынан айырмашылығы тек кіші разрядтарда ғана болады. Төмендегі мысалда қайталанатын цифрлар бір символмен (а символы) алмастырылған.

5. **Графиктік ақпаратты сызу.**
- «0» мен «1» серияларын кодтау.

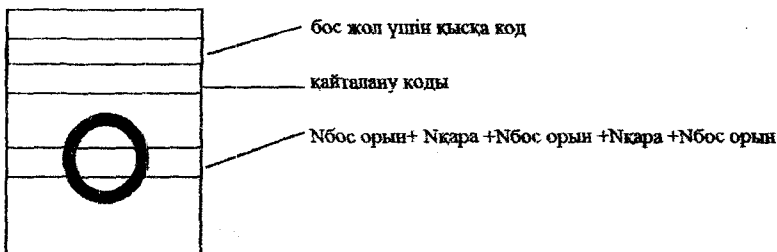
| | |
|---------|---------|
| 5538142 | 5538142 |
| 5538143 | a3 |
| 5538145 | a5 |
| 5538161 | a61 |
| 5538163 | a3 |

| | | |
|---------------------|-----|---|
| 0 | 000 | Мысалы: нөлдер саны 18 болса, код – 111 010 болады. |
| 00 | 001 | |
| 000 | 010 | |
| 0000 | 011 | |
| 00000 | 100 | |
| 5-тен артық нөлдер | 101 | |
| 10-нан артық нөлдер | 110 | |
| 15-тен артық нөлдер | 111 | |

- Қайталанатын символдарды олардың санымен алмастыру.

Мысалы: AAAABBBCCCCDD үшін код – 4A3B5C2D болады.

– Қысқа кодтар және қайталанатын кодтар:



3.2.2 Деректерді сығу әдістері мен құралдары

Деректерді сығу алгоритмдерін бағалау өлшемдеріне:

– *сығу сапасы (коэффициенті немесе дәрежесі)*, бұл сығылған деректер ұзындығының (бит) деректердің бастапқы ұзындығына қатынасына тең;

– *кодтау мен декодтау жылдамдығы*, бұл деректерді кодтау мен декодтауға жұмсалатын уақытпен анықталады;

– *қажет етілетін жад көлемі* жатады.

Деректерді сығу әдістері мен құралдары (megabook.ru мегаэнциклопедия материалдары және [23], [24] бойынша):

– *Хаффман сығу әдісі* (Huffman compression method, CCITT кодтауы) – 1952 жылы Дэвид Хаффман (David Huffman) ашқан, 2.4-бөлімде бұл кодтау әдісі қарастырылған. Телефония мен телеграфия бойынша халықаралық консультациялық комитет (CCITT) осы код негізінде қара-ақ түсті суреттерді телефон арналары және деректерді жіберу желілері арқылы факспен жіберуге арналған бірнеше коммуникациялық хаттамалар әзірледі (Т.4 CCITT пен Т.6 CCITT стандарттары немесе CCITT group 3 сығуы мен CCITT group 4 сығуы).

– *Фрактальдық сығу* (fractal compression) – растрлық суреттерді фракталдарға түрлендіруге негізделген сығу әдісі. Бейнені фракталь түрінде сақтау пиксель түрінде сақтағанға қарағанда дискілік жадты 4 есе аз қажет етеді.

– *ART* – мәтінді, графиканы, аудио мен видеоны сығу әдісі, суретті талдауға мен олардың түс, бөгеуілдер, шеті, қайталанатын элементтері сияқты басты белгілерін анықтауға негізделген.

– *AC3 Dolby* – аудиодеректерді бір файлға сығып, сақтап, 32 кбит/с бастап 640 кбит/с дейін жылдамдықпен жіберуге арналған сығудың әдісі мен форматы.

– *DJVU* (DjVu, djvu, deja vu) – мәтін және суреттерден тұратын басылымның сканерленген беттерін динамикалық сығуға арналған технология мен формат.

– *DVI* (Digital Video Interactive) – цифрлық түрдегі аудио, бейне (видео) ақпаратты динамикалық сығу және қалпына келтірудің бұл жүйесі CD-ROM-ға толық форматтағы бейнефильмдерді дыбысымен жазуға мүмкіндік береді.

– *EAD* (Encoded Archival Description) – кодтау стандарты (АҚШ Конгресі кітапханасының Network Development and MARC Standards Office бөлімшесі Society of American Archivists-пен бірлесіп 1998 жылы жасаған, 2002 жылы жаңартылған), стандартта архивтік және кітапханалық іздеу құралдарына (finding aids) арналған кодтау схемасын құру, әзірлеу, қолдау қағидалары айқындалған.

– *Image compression manager* – суреттің динамикалық сығылуын басқару программасы, суреттерді сығудың, қалпына келтірудің әртүрлі әдістерін (MPEG, JPEG) қолдануға мүмкіндік береді.

– *JBIG* (Joint Bi-level Image Experts Group) – екі түсті (екі деңгейлік) суреттерді шығынсыз сығу әдісі, 1988 жылы ISO мен CCIT-тің екі деңгейлі суреттер бойынша сарапшыларының біріккен тобы жасады. JBIG әдісі 1993 жылы екі деңгейлі деректердің кодтау стандарты болып жарияланды, оған дейінгі MR (Modified READ) және MMR (Modified Modified READ) сығу алгоритмдерінің тиімділігі аз болды.

– *LZW* (Lempel-Ziv-Welch) – динамикалық сығу әдісі, бірегей деректер тізбектеріне қысқа маркерлер (кілттер) меншіктеледі, фразалар деп аталатын бірдей деректер тізбектерін бүкіл файлдан іздеуге және сөздіктерде сақтауға негізделген. Бұл әдіс 3.4-бөлімде қарастырылады.

– *MP3* (Moving Pictures Experts Group, Layer 3) – динамикалық сығу әдісі және аудиодеректер файлын жазудың арнайы

форматы. Дыбыстық деректерді сығу дәрежесі жоғары, мультимедиа қолданбаларында, соның ішінде цифрлық ойнатқыштарда (player), Интернетте қолданылады.

– *RLE* (Run Length Encoding) – қайталанатын символдар жолдарының физикалық өлшемдерін қысқартуға негізделген графикалық деректерді динамикалық сығу әдісі.

3.2.3 Сығу алгоритмдерінің жіктелуі

Символдар статистикасын қолдануына байланысты деректерді сығу алгоритмдері 3 топқа бөлінеді [5, Вернер]:

1. Статикалық алгоритмдер. Статикалық алгоритмдерде жиілік кестесі мәтін бойынша құрылмайды, алдын-ала есептелген дайын кесте қолданылады, мәтінді бір рет қана қарастырады. Оның мысалына статикалық Хаффман кодтауы жатады. Неміс тіліндегі әдеби мәтінді Хаффман әдісімен сыққанда нәтижесі стандартты ASCII кодтауын қолданып сығуға қарағанда 50 пайыз жоғары болған. Архиваторларда, форматтарда (*JPEG, ARJ*) қолданылады.

2. Адаптивті алгоритмдер. Мұның мысалына Хаффманның модификацияланған кодтауы жатады. Шығынсыз сығу алгоритмі болып табылады. Бұл кодтау әдісінде символдардың ықтималдықтарының үлестіруі алдымен бірқалыпты болады, содан кейін статистика жинақталуына қарай өзгеріп отырады.

3. Динамикалық алгоритмдер. Динамикалық алгоритмдер мәтінді 2 рет қарастырады, алдымен символдардың мәтіндегі кездесу жиіліктерінің кестесін құру үшін, одан соң – кодтау үшін, яғни бастапқы символдарды кодпен алмастыру үшін. Оның мысалына динамикалық Хаффман кодтауы жатады. Архиваторларда, форматтарда (*UNIX, ICE* үшін *compact* утилиті), *ITU V.42.bis* ұсынысында қолданылады.

Символдар статистикасы өте жиі жағдайда алдын-ала белгісіз болады, ал тиімді кодтау үшін жиіліктік талдау қажет. Бұл жағдайда **эмбебап сығу алгоритмдері** қолданылады. Эмбебап алгоритмдер түп мәнісі адаптивті болғандықтан априорлы статистиканы керек қылмайды. Мұнда тиімді кодтау ақпарат кодерге енгеннен кейін бірден басталады. Ұсынылған

алгоритмдердің әрқайсысы жоғары сығу дәрежесін алуға мүмкіндік береді.

Әмбебап сығу әдістерінің ішінде 2 маңызды алгоритмдер бар:

1) *арифметикалық кодтау*, динамикалық жиіліктік талдауға негізделген, шығынсыз сығу алгоритмі. Қолданысы: JPEG, RAR;

2) *Лемпель-Зив алгоритмдері*. Бұл әмбебап Лемпель-Зив алгоритмдері шығынсыз сығу алгоритміне жатады, сөздіктер әдісіне негізделген, фразалар сөздігін қолданып, кодтау мен декодтауда алдыңғы кездескен ақпаратты ескереді. Қазіргі таңда Лемпель-Зив алгоритмдері ZIP, RAR архиваторларында, модемдік байланыс стандарттарында, графикалық файл форматтарында қолданылады.

Лемпель-Зив алгоритмдерінің бірнеше нұсқалары бар:

– 1977, 1977 жылдары израилдық математиктер Авраам Лемпель мен Якоб Зив LZ77, LZ78 (Лемпель-Зив) алгоритмдерін жасады.

– 1982 жылы бұл алгоритмдерді Сторер мен Шиманский жетілдіріп, LZSS (Лемпель-Зив-Сторер-Шиманский) алгоритмін ұсынды.

– 1984 жылы Уелч жетілдірген алгоритм LZW (Лемпель-Зив-Уелч) алгоритмі деп аталады.

3.3 Арифметикалық кодтау

Арифметикалық кодтау XX ғасырдың 70-ші жылдарында пайда болды, энтропияға негізделген, ақпаратты шығынсыз сығу алгоритміне жатады. Хаффманның кодтау алгоритмі хабардың әрбір символына 1 биттен артық емес ақпарат жібере алмайды. Нөл мен бірден тұратын хабарда бірліктер нөлге қарағанда 10 есе жиі кездеседі деп қарастырсақ, Хаффманның әдісі арқылы кодтағанда 0 үшін де, 1 үшін де бірдей артық бит жұмсалады. Хаффманның әдісі хабардың бір символындағы биттің орташа санының минимальды мәніне 1 битті береді. Кейбір символдарды бірден аз битпен кодтауға мүмкіндік беретін кодтау схемаларының арасындағы ең үздігі – арифметикалық кодтаудыкі. Арифметикалық кодтауда жасалған кодтық сөздің ұзындығы бастапқы мәтіннің энтропиясынан аспайды. Хаффманның кодтау алгоритміне

қарағанда ықтималдықтары тең қалыпты емес түрде үлестірілген символдардан тұратын мәтінді сығу нәтижесі жоғары. Кодтау кезінде мәтінге $[0,1)$ кесіндісі аралығынан нақты сан сәйкес қойылады.

Арифметикалық кодтау әдісінде ықтималдықтың үлестірімі бойынша кодтау үшін тандалған дискретті кездейсоқ шама үшін, дискретті кездейсоқ шаманың әр мәні үшін шекаралық нүктелерде қиылысатын кесіндіден тұратын кесте құрылады. Осы кесінділердің қосындысы $[0,1)$ кесіндіні құрауы қажет, ал олардың ұзындығы дискретті кездейсоқ шаманың мәніне сәйкес ықтималдылықтарға пропорционалды болуы тиіс.

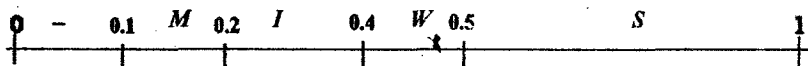
Кодтау алгоритмі дискретті кездейсоқ шама мәнінің кезектілігін ашықтайтын кесіндіні құрудан тұрады. Одан кейін кесінді құру үшін, оның ішкі бөлігінде жататын және минимальды мүмкін болатын екінің оң, бүтін дәрежесіне бөлінген, бүтін санға тең санды табамыз. Бұл сан қарастырылып отырған кезектіліктің коды болып табылады. Барлық нақты болатын мүмкін кодтар – бұл нөлден үлкен және бірден кіші сандар, сондықтан алдыдағы нөл мен ондық нүктені алып тастауға болады, бірақ хабардың соңын белгілейтін тағы да бір арнайы код-маркер керек. Кесінділер былай құрылады: егер $n-1$ ұзындықтағы хабар үшін кесінді бар болса, онда n ұзындықтағы хабар үшін кесіндіні құрастыру үшін, оны қарастырып отырған дискретті кездейсоқ шаманың мәні қанша болса, сонша бөлікке бөлеміз. Бұл бөлу біріншідегідей (реттілікті сақтаумен) жүзеге асырылады. Одан кейін алынған кесінділерден берілген нақты n ұзындықтағы кезектілікке сәйкес келетінін таңдаймыз.

Алдында қарастырылған әдістерден бұл *арифметикалық кодтаудың негізгі ерекшелігі* – оның үздіксіздігінде, яғни блоктауды қажет етпейтіндігінде. Арифметикалық кодтаудың тиімділігі сығылатын хабардың ұзындығы артқан сайын арта түседі (бұл Хаффман және Шеннон–Фано кодтауында болмайды). Арифметикалық кодтау Хаффман кодтауына қарағанда жақсы сығады, бірақ ол тәжірибеде сирек қолданылады, өйткені ол кейінірек пайда болды және үлкен есептеу ресурстарын қажет етеді.

Берілген деректерді (мысалы, файлды) сығуда барлық қарастырылған әдістер екі тәсілді талап етеді. *Біріншісі* – символ ықтималдылығының жақын мәндері ретінде қолданылатын

символдардың жиілігін жинақтау және *екіншісі* – сығылатын жолдың символдарының кездесу жиілігі (мысалы: SWISS_MISS жолы үшін S – 0.5, W – 0.1, I – 0.2, M – 0.1, _ – 0.1).

Арифметикалық кодерде әрбір символ [0, 1) диапазон аралығында кездеседі. Осы мысалға сәйкес әліпби символдары үшін келесі аралықтар (интервалдар) жиыны алынады (3.1-сурет).



3.1-сурет. Символдарды бейнелейтін аралықтардың орналасуы

Кодтауы. Кодтау үшін мәтін мен ондағы символдардың кездесу жиіліктері беріледі.

1) 3.1-суреттегі сияқты [0, 1) кесіндісін қарастырып, берілген мәтіннің әрбір символына ұзындығы оның жиілігіне тең кесіндіні сәйкес қоямыз.

2) Кезекті символды мәтіннен оқып, оған сәйкес кесіндіні қараймыз. Осы қарастырып отырған кесіндіні символдардың жиіліктеріне пропорционал етіп сәйкес бөліктерге бөлеміз.

3) Мәтін аяқталғанша 2-ші қадамды қайталаймыз.

4) Соңғы шыққан кесіндіден кез келген санды аламыз, бұл арифметикалық кодтау нәтижесі болады.

Арифметикалық кодтаудың мәнін түсіну үшін келесі мысалды қарастырайық. Кірістік ағынның үш символын кодтау керек болсын.

Арифметикалық кодтаудың мысалы. X дискретті кездейсоқ шамасы тек 0 және 1 мәнін $2/3$ және $1/3$ ықтималдықпен қабылдасын делік. 0 мәнін $[0; 2/3]$; ал 1 мәнін $[2/3; 1]$ кесіндісіне орналастырамыз. Онда мына \bar{X} дискретті кездейсоқ шамасы үшін

$$\dim(\bar{X}) = 3, H\bar{X} = \frac{H\bar{X}}{3} = \log_2 3 - 2/3 \approx 0.9183 \text{ бит/символ}$$

арифметикалық кодты құрудың кестесі төменде берілген (3.3-кесте). (Мысал В.В.Лидовский, Основы теории информации и криптографии [3] әдебиетінен алынды)

Кесте 3.3 – \bar{X} дискретті кездейсоқ шамасы үшін арифметикалық кодты құру

| Аралықтар мен кодтар | Ықтималдық | Хаффман коды |
|---|------------|--------------|
| $111 \left[\frac{26}{27}, 1 \right] \in \frac{31}{32} = 0.11111$ | 1/27 | 0000 |
| $11 \left[\frac{8}{9}, 1 \right] \quad 110 \left[\frac{8}{9}, \frac{26}{27} \right] \in \frac{15}{16} = 0.1111$ | 2/27 | 0001 |
| $101 \left[\frac{22}{27}, \frac{8}{9} \right] \in \frac{7}{8} = 0.111$ | 2/27 | 010 |
| $1 \left[\frac{2}{3}, 1 \right] \quad 10 \left[\frac{2}{3}, \frac{8}{9} \right] \quad 100 \left[\frac{2}{3}, \frac{22}{27} \right] \in \frac{3}{4} = 0.11$ | 4/27 | 001 |
| $011 \left[\frac{16}{27}, \frac{2}{3} \right] \in \frac{5}{8} = 0.101$ | 2/27 | 011 |
| $01 \left[\frac{4}{9}, \frac{2}{3} \right] \quad 010 \left[\frac{4}{9}, \frac{16}{27} \right] \in \frac{1}{2} = 0.1$ | 4/27 | 100 |
| $001 \left[\frac{8}{27}, \frac{4}{9} \right] \in \frac{3}{8} = 0.011$ | 4/27 | 101 |
| $0 \left[0, \frac{2}{3} \right] \quad 00 \left[0, \frac{4}{9} \right] \quad 000 \left[0, \frac{8}{27} \right] \in \frac{1}{4} = 0.01$ | 8/27 | 11 |

$$ML_1(\bar{X}) = \frac{65}{81} \approx 0.8025 \text{бит/сим. (арифметикалық)}$$

$$ML_1(\bar{X}) = 76/81 \approx 0.9383 \text{бит/сим (блоктық Хаффман)}$$

$$ML_1(X) = ML(X) = 1 \text{бит/сим (Хаффман)}$$

Арифметикалық кодтау үшін ақпарат бірлігіндегі биттің орташа саны энтропияға қарағанда аз болды. Бұл қарастырылып отырған жай ғана кодтау схемасында ақпарат соңындағы код-маркер сипатталмаған, егер оны енгізсек, онда ол биттің орташа санын энтропиядан көп етеді.

Декодтауы. Арифметикалық кодпен кодталған хабардан бастапқы ақпаратты алу (декодтау) келесі алгоритм бойынша жүзеге асады.

1-қадам. Ұзындықтары мәтіндегі символдардың кездесу жиілігіне тең етіп бөліктерге бөлінген $[0, 1)$ кесіндісінен кодтық сөзге, яғни нақты санға сәйкес келетін ішкі кесінді (немесе дискретті кездейсоқ шама мәнін кодтау кестесінде ағымдағы коды

бар аралық) анықталады, осы ішкі кесіндіге (аралыққа) сәйкес символды бастапқы ақпараттың кезекті бір символы ретінде жауапқа жазамыз. Егер бұл символ ақпарат соңының маркері болса, онда бұл соңы болады.

2-қадам. Ішкі кесінді мен нақты санды нормалаймыз. Ағымдағы кодтан оның аралығы бар төменгі шекара алынады, оны осы аралықтың ұзындығына бөлеміз. Алынған сан кодтың жаңа ағымдағы мәні болып саналады. 1 қадамға ауысамыз. Осылайша бастапқы хабар толық қапшына келгенше (файл соңына жеткенше) жалғастырамыз.

Мысал. (Witten I., Neal R., Cleary J. Arithmetic Coding For Data Compression Communications of the ACM, vol. 30, no. 6, pp.520-540, June 1987. материалы бойынша)

Бастапқы хабар ретінде «ea!i!» мәтіні, әліпби символдары {a,e,i,o,u,!} және сәйкесінше осы символдардың ықтималдықтары «a»-0.2, «e» - 0.3, «i» - 0.1, «o» - 0.2, «u» - 0.1, «!» - 0.1 берілсін.

1-қадам. Бастапқыда мәтінге сәйкес [0; 1) кесіндісі қарастырылады. Әліпби символдарының ықтималдықтарына сәйкес кесінді 3.3-кестеде көрсетілгендей бөліктерге (интервалдарға) бөлінеді. Кезекті символды кодтау үшін 3.2-суретте көрсетілгендей қарастырылып отырған символға ішкі кесіндіні анықтап, осы кесіндіні символдардың ықтималдықтарына сәйкес бөліктерге бөлеміз.

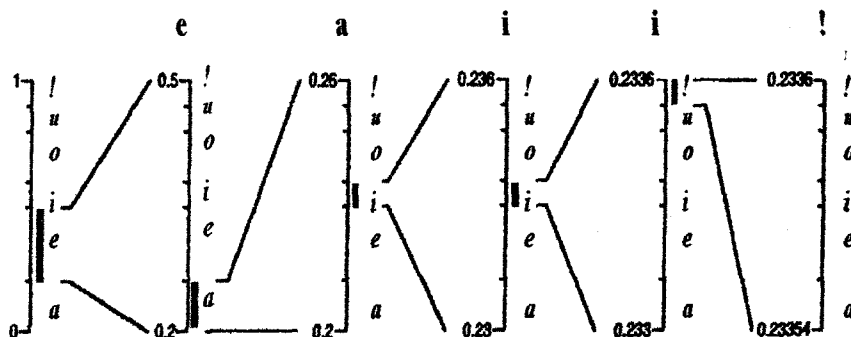
Кесте 3.3 – [0; 1) кесіндісінің { a,e,i,o,u,! } әліпби символдарының ықтималдықтарына сәйкес бөліктерге бөлінуі

| Символ | Ықтималдық | Интервал |
|--------|------------|------------|
| a | 0.2 | [0.0; 0.2) |
| e | 0.3 | [0.2; 0.5) |
| i | 0.1 | [0.5; 0.6) |
| o | 0.2 | [0.6; 0.8) |
| u | 0.1 | [0.8; 0.9) |
| ! | 0.1 | [0.9; 1.0) |

2-қадам. Кодер мен декодерге бастапқы кесіндінің [0; 1) екені белгілі. Кодталатын «ea!i!» мәтініндегі 1-ші «e» символын қарастырған соң, кодер интервалды осы символға сәйкес келетін ішкі кесіндіге [0.2; 0.5) дейін қысқартады. 2-ші символ «a» бұл

аралықты бестен бір бөлігінің біріншісіне дейін қысқартады, өйткені «а» үшін $[0.0; 0.2)$ интервал берілген, нәтижесінде кезекті жұмыстық интервалды $[0.2; 0.26)$ аламыз, алдыңғы интервал ені 0.3 бірлік, ал оның бестен бір бөлігі 0.06 болады. Келесі символ «і» үшін сәйкес интервал $[0.5; 0.6)$, оған сәйкес жұмыстық интервал $[0.2; 0.26)$ оны $[0.23, 0.236)$ интервалына дейін қысқартады. Осылай жалғастыра отырып нәтижесінде арифметикалық кодты аламыз:

Бастапқыда – $[0.0; 1.0)$
 "е" символын оқыған соң – $[0.2; 0.5)$
 "а" символын оқыған соң – $[0.2; 0.26)$
 "і" символын оқыған соң – $[0.23; 0.236)$
 "і" символын оқыған соң – $[0.233; 0.2336)$
 "!" символын оқыған соң – $[0.23354; 0.2336)$



3.2-сурет. Кезекті символға кесіндіні анықтау, арифметикалық кодты алу

Декодтау мысалы. Декодер мәтін туралы және соңғы интервал $[0.23354; 0.2336)$ екенін білсін делік. Декодталатын символдың "е" болатынын біледі, өйткені соңғы интервал 3.3-кестеде анықталған осы символға бөлінген интервалда жатыр. Декодер жұмысы мына ретпен қайталанады: Бастапқы кесінді $[0.0; 1.0)$, ал "е" символы анықталған соң кезекті кесінді $[0.2; 0.5)$ болады. Бұдан келесі символ "а" болатыны анықталады, өйткені бұл $[0.2; 0.26)$ интервалына әкеледі, оның құрамына кодтық сөзге сәйкес келетін соңғы интервал $[0.23354; 0.2336)$ кіріп тұр. Осылайша жалғастыра отырып декодер бүкіл мәтінді декодтайды. Алайда, декодтау

процесін аяқтау үшін декодерге мәтін соңын дәл уақытында білу қажет, сондықтан әр мәтінді кодер де, декодер де білетін EOF (файл соңы) символымен аяқтайды, оны 3.3-кестеде анықталған "!" символымен белгілейді. Декодер "!" символын кезіктіргенде жұмысын тоқтатады.

Ескерту. Арифметикалық кодтау туралы, кодтың программалық жүзеге асуы туралы мына сайттан <http://compression.ru/download/ar.html> және [3]- [5] әдебиеттерден танысуға болады.

3.3.1 Адаптивті арифметикалық кодтау

Арифметикалық кодтау үшін де, Хаффман әдісі арқылы кодтау үшін де адаптивті алгоритмдер бар.

Берілген жиындағы символдар кезектілігі үшін арифметикалық кодты құрастыруды келесі алгоритм бойынша жүзеге асыруға болады. Әр символға оның салмағы сәйкестендіріледі: алғашқыда ол бәрі үшін 1-ге тең. Барлық символдар нақты ретпен (мысалы, өсуі бойынша) орналасады. Әрбір символдың ықтималдылығын өз салмағын барлық символдардың салмағына бөліп анықтаймыз (3.2-сурет). Кезектегі символды алғаннан соң және ол үшін аралықты жазғаннан кейін, осы символдың салмағы 1-ге артады (салмақты кез-келген реттегіш тәсілмен өсіруге болады).

Берілген символдар жиыны – бұл ASCII+ кодының символдары болсын. Сығылатын хабардың басында ашу алгоритмін тоқтатуды қамтамасыз ету үшін оның ұзындығын беру керек немесе хабар соңына қосымша символ-маркер енгізу қажет. Егер сығу үшін файл форматын білсек, онда салмақты алғашқы бірқалыпты үлестірудің орнына осы білімді есептей отырып үлестіруді таңдауға болады. Мысалы, мәтіндік файлда басқарушы символдар тізбегі болмау керек және олардың салмағын нөлге теңестіруге болады.

Кодтау мысалы. Берілген әліпбидің символдары – А, В, С, ал сығылатын хабар АССВСАААВС болсын. Хабар соңының белгісін (маркерін) енгіземіз. Берілген алгоритмге сәйкес кодтауды 3.3-ші суретте көрсетілген схема бойынша жасауға болады. Соның нәтижесінде:

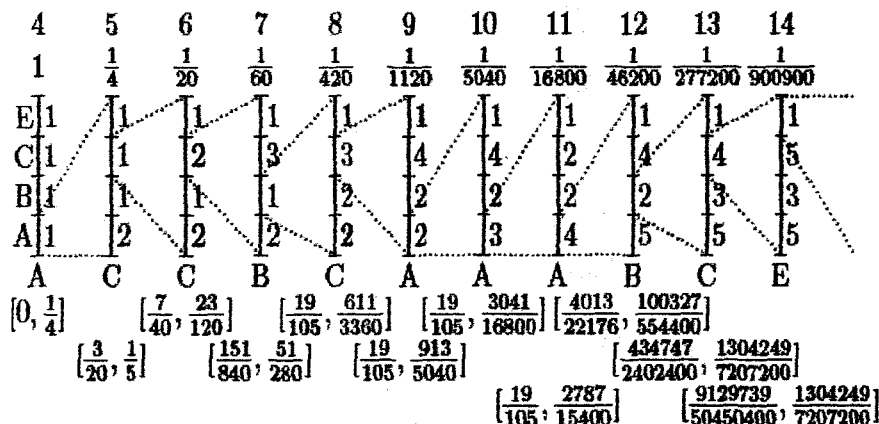
759021

$$2^{22} = 4194304$$

$$= 0.0010111001010011101101_2 \epsilon \left(\frac{9129739 \ 1304249}{50450400 \ 7207200} \right)$$

$$\text{code}(\text{ACCBVCAAABC}) = 0010111001010011101101$$

Сондықтан, $L_1(\text{ACCBVCAAABC}) = 2.2$ бит/символ. Хаффман адаптивті алгоритмімен алынған нәтиже 4,1 бит/символ, бірақ, егер әріптерді 8 битпен емес, 2 битпен кодтасақ, онда нәтиже 2,3 бит/символ болады. Схеманың бірінші қатарында символдардың салмақтарының қосындылары, ал екіншісінде ағымдағы кесінділердің ұзындығы жазылған.



3.3-сурет. Символдар кезектілігі үшін арифметикалық код

Адаптивті арифметикалық кодты декодтау. Адаптивті арифметикалық кодты декодтау тәсілі келтірілген адаптивті емес тәсілмен ұқсас. *Айырмашылығы:* екінші қадамда жаңа кодты алған соң символдар салмағын жаңадан үлестіруге сәйкес ортақ кесіндіні бөлуді қайта құрастыру керек. Хабар соңынан және басынан берілген маркерімен символдар санын алу жұмыстың аяқталғанын білдіреді.

Декодтау мысалы. Хабардың символдар жиыны А, В, С, Е, соңғы символ Е – хабар соңының маркері болсын.

0010111001010011101101 коддын 3.4-кестеде көрсетілгендей ашамыз.

$$0.0010111001010011101101_2 = \frac{759021}{4194304}$$

Кесте 3.4 – Арифметикалық кодты декодтау

| Салмақтары | | | | Код саны және оның интервалы | Символ | Интервал ұзындығы |
|------------|---|---|---|---|--------|-------------------|
| A | B | C | E | | | |
| 1 | 1 | 1 | 1 | $\frac{759021}{4194304} \in \left(0, \frac{1}{4}\right)$ | A | 1/4 |
| 2 | 1 | 1 | 1 | $\frac{759021}{1048576} \in \left(\frac{3}{5}, \frac{4}{5}\right)$ | C | 1/5 |
| 2 | 1 | 2 | 1 | $\frac{649377}{1048576} \in \left(\frac{1}{2}, \frac{5}{6}\right)$ | C | 1/3 |
| 2 | 1 | 3 | 1 | $\frac{375267}{1048576} \in \left(\frac{2}{7}, \frac{3}{7}\right)$ | B | 1/7 |
| 2 | 2 | 3 | 1 | $\frac{529717}{1048576} \in \left(\frac{1}{2}, \frac{7}{8}\right)$ | C | 3/8 |
| 2 | 2 | 4 | 1 | $\frac{5429}{393216} \in \left(0, \frac{2}{9}\right)$ | A | 2/9 |
| 3 | 2 | 4 | 1 | $\frac{16287}{262144} \in (0, 0.3)$ | A | 0.3 |
| 4 | 2 | 4 | 1 | $\frac{27145}{131072} \in \left(0, \frac{4}{11}\right)$ | A | 4/11 |
| 5 | 2 | 4 | 1 | $\frac{298595}{524288} \in \left(\frac{5}{12}, \frac{7}{12}\right)$ | B | 1/6 |
| 5 | 3 | 4 | 1 | $\frac{240425}{262144} \in \left(\frac{8}{134}, \frac{12}{13}\right)$ | C | 4/13 |
| 5 | 3 | 5 | 1 | $\frac{1028373}{1048576} \in \left(\frac{13}{14}, 1\right)$ | E | |

Жалпы адаптивті мен адаптивті емес арифметикалық кодтаулар ұқсас, олардың айырмашылығы: екінші қадамда жаңа кодты алған соң символдар салмағын жаңадан үлестіруге сәйкес ортақ кесіндіні бөлуді қайта құрастыру керек.

3.4 Лемпель-Зив кодтары

Шеннон–Фано әдісі, Хаффман кодын және арифметикалық кодтауды жалпы түрде символдардың кездесу ықтималдықтарын, яғни хабардағы кездесу жиіліктерін қолданатындықтан *статистикалық әдістер* деп атайды. Лемпель-Зив кодтары кодтау үшін сөздіктерді қолданатындықтан *сөздіктік алгоритмдерге* жатады, олар практикада кеңінен қолданылады.

Лемпель-Зив коды мен оның нұсқаларын қолданып бастапқы көлемі 1832 бит болатын хабарды сығу нәтижелері төмендегі 3.5-кестеде берілген. Кестеден LZ78 сығу алгоритмінің тиімді екені анық байқалып тұр.

Кесте 3.5 – Лемпель-Зив коды мен оның нұсқаларының мәтінді сығу көрсеткіштері

| | LZ77 | LZ78 | LZSS | LZW |
|-------------------------------------|----------|----------|----------|----------|
| Мәтіннің бастапқы көлемі | 1832 бит | 1832 бит | 1832 бит | 1832 бит |
| Мәтіннің сығылғаннан кейінгі көлемі | 1740 бит | 1308 бит | 1382 бит | 1395 бит |
| Сығу дәрежесі | 5.02 % | 28.6% | 24.56% | 23.85% |

Ескерту. Лемпель-Зив кодтары мен нұсқалары туралы толығырақ мына ақпарат көздерін қараңыз: [3-4], [21] әдебиеттер және Ливак Е.Н. Алгоритмы сжатия. <http://mf.grsu.by/UchProc/livak/po/comprsite/>.

3.4.1 LZ77 (Лемпель-Зив) алгоритмі

LZ77 кодының алгоритмінің танымал болуына практикалық маңыздылығының жоғары болуы, сығу тиімділігінің жоғары, аса қарапайым болуы себепкер болды. LZ77 хабардың қаралған бөлігін сөздік ретінде қолданады. Мәтінді сығу үшін хабардың кезекті қарастырылатын бөлігін сөздікке жасалған көрсеткішпен алмастырады. Алгоритм екі бөліктен тұратын терезені қолданады. Өлшемі үлкен бөлікке, яғни сөздікке хабардың қарастырылған бөлігі жазылады. Ал өлшемі кіші бөлігі буфер ретінде қолданылады, оған кодталмаған символдар жазылады. Әдетте терезе бірнеше килобайттан тұрса, ал буфердің көлемі жүз байттан аспайды. Алгоритм сөздіктен (бірінші терезеден) буфердің мазмұнымен сәйкес келетін бөлікті (яғни, буфердегі жазылғанмен бірдей болатын фрагментті) сөздіктен іздейді.

Кодтау барысында LZ77 алгоритмі 3 элементтен тұратын кодты жасайды, яғни өңделетін мәтінді [a, n, z] көрсеткіштер тізбегіне айналдырады, мұндағы:

a – сөздіктегі фразаның (ішкі жолдың) салыстырмалы адресі, яғни, буфердегі жазылғанға сәйкес келетін сөздіктегі фразаның жол басына қатысты ығысуы; басқаша айтқанда буфердің алғашқы жолының сөздіктегі бастапқы жолға қатысты қысқаруы;

n – сөздіктегі сәйкес келген фразаның (жолдың) ұзындығы;

z – сөздіктегі фразадан (жолдан) кейін тұрған буфердегі өңделетін мәтіннің келесі бірінші символы.

Кодтау сөздіктегі фразаларға көрсеткіштер арқылы жүреді. Содан кейін алгоритм терезені сәйкес келген фразаның ұзындығы+1 символға солға жылжытады.

Егер буфердегіге сөздіктен ешқандай сәйкес келетін фраза табылмаса, алгоритм <0, 0, буфердегі 1-ші символ> кодын береді де жұмысын жалғастырады. Нәтижесінде, бұл жағдайда тиімсіз болса да, алгоритм кез келген хабарды кодтай алады.

Кодтау мысалы (мысал оқу құралы соңында көрсетілген әдебиеттен алынды). Терезенің өлшемі 20 символ, сөздіктікі – 12 символ, ал буфер өлшемі 8 символ. Кодталатын ақпарат ПРОГРАММНЫЕ ПРОДУКТЫ болсын. Сөздікті толық деп есептейік. Сөздікте «ПРОГРАММНЫЕ» жолы, ал буферде «ПРОДУКТЫ» жолы бар. Алгоритм сөздікті қарастыра отырып «Про» фразасын табады, ол сөздіктің басында болғандықын ығысуы –0, ал фраза ұзындығы – 3 және буфердегі осы фразадан кейінгі келесі символ Д болады. Осыған байланысты жасалған код (0, 3, Д) тізбегі болады. Осыдан кейін алгоритм барлық сөздіктегі мәтінді 4 орынға солға жылжытады, себебі үшеуі табылған ПРО сөзіне, біреуі Д символына байланысты. Келесі қадамда сөздіктен РАММНЫЕ ПРОД жолы қарастырылады, ал буферде УКТЫ сөзін қалды. Сәйкес фраза табылмағандықтан, буфердегінің бірінші әрпін У-ды көрсетіп (0, 0, У) деп кодын береді де, сөздіктегі мәтінді бір символға солға жылжытады. Осылай жалғастыра береді

| Сөздік (12 символ) | Буфер (8 символ) | LZ77 коды [a, n, z] |
|--------------------|------------------|---------------------|
| ← ПРОГРАММНЫЕ ← | ПРОДУКТЫ | (0, 3, Д) |
| РАММНЫЕ ПРОД | УКТЫ | (0, 0, У) |
| | ... | |

Алгоритмнің сөздікке негізделгеніне байланысты мұндай кодтауды сөздіктік кодтау деп атайды. Сөздіктік кодтарды декодтау қарапайым, кері ретпен жүреді.

LZ77 алгоритмімен кодтау. Мысал. 3.6-кестеде КРАСНАЯ КРАСКА мәтінін LZ77 алгоритмімен кодтаудың мысалы көрсетілген (мысал [3] әдебиеттен алынды). Соңғы жолдағы А әрпі сөздіктен алынбайды, өйткені ол – соңғы әріп.

Екілік кодтың ұзындығы: $n = N \cdot \{[\log_2 n_c]^* + [\log_2 n_6 + 1]^* + 8\}$ бит, мұндағы n_c – сөздіктің ұзындығы ($n_c = 8$), n_6 – буфердің ұзындығы ($n_6 = 5$), ал $[\]^*$ – санды үлкен жағына қарай дөңгелектеу операциясы.

Символдар 8 биттік ASCII+ кодымен кодталған деп саналады.

Жол саны $N = 9$. Бұдан КРАСНАЯ КРАСКА мәтінінің LZ77 алгоритмімен жасалған кодының ұзындығы:

$$n = 9 \cdot \{[\log_2 8]^* + [\log_2 (5+1) + 1]^* + 8\} = 9 (3+3+8) = 126 \text{ бит}$$

Кесте 3.6 – LZ77 алгоритмімен кодтау мысалы

| Сөздік (8) | Буфер (5) | LZ77 коды [a, n, z] |
|------------|-----------|---------------------|
| «.....» | «КРАСН» | <0, 0, К> |
| «.....К» | «РАСНА» | <0, 0, Р> |
| «.....КР» | «АСНАЯ» | <0, 0, А> |
| «....КРА» | «СНАЯ» | <0, 0, С> |
| «...КРАС» | «НАЯК» | <0, 0, Н> |
| «.КРАСН» | «АЯКР» | <5, 1, Я> |
| «.КРАСНАЯ» | «КРАС» | <0, 0, ''> |
| «КРАСНАЯ» | «КРАСК» | <0, 4, К> |
| «АЯКРАСК» | «А....» | <0, 0, А> |

LZ77 кодын декодтау. LZ77 кодын декодтау оларды жасаудан әлдеқайда жеңіл болып табылады, себебі, сөздікте іздеу жүргізілмейді.

Мысал. LZ77 алгоритмімен сығылған хабардың коды <0,0, «К»> <0,0, «Р»> <0,0, «А»> <0,0, «С»> <0,0, «Н»> <5,1, «Я»> <0,0, «»> <0,0, «К»> <0,4, «К»> <0,0, «А»> берілсін.

Сөздіктің ұзындығы – 8 символ.

Төмендегі 3.7-кестеде LZ77 алгоритмімен сығылған хабарды декодтаудың мысалы көрсетілген. (Мысал [3] әдебиеттен алынды)

Кесте 3.7 – LZ77 алгоритмімен сығылған хабарды декодтау

| Бастапқы код | Баспа | Сөздік |
|--------------|---------|--------------|
| 0«К» | «К» | «..... К» |
| 0«Р» | «Р» | «..... КР» |
| 0«А» | «А» | «.... КРА» |
| 0«С» | «С» | «... КРАС» |
| 0«Н» | «Н» | «... КРАСН» |
| 1<5,1> | «А» | «.. КРАСНА» |
| 0«Я» | «Я» | «.. КРАСНАЯ» |
| 0« » | « » | «КРАСНАЯ » |
| 1<0,4> | «КРАС » | «НАЯ КРАС» |
| 1<4,1> | «К» | «АЯ КРАСК» |
| 1<0,1> | «А» | «Я КРАСКА» |

LZ77 алгоритмінің артықшылықтары:

- Жиі кездесетін символдардың тізбегі тиімді кодталады.
- Жиі қайталанатын символдар тиімді кодталады.
- «Нөлдік фразаларды» кодтау үшін көп бит саны жұмсалады.
- Сирек кездесетін символдар мен символдардың тізбегі уақыт өткен соң фразалар сөздігінен жойылады.
- Ұзын мәтіндердегі деректерді сығу дәрежесі 50-60% жетеді.
- Өте ұзын мәтіндерді сығуда бұл әдіс тиімді, өйткені артықтық жоғалады, бір символды кодтауға қажетті тбиттердің орташа саны осы мәтіннің энтропиясына жуық болады.

LZ77 алгоритмінің кемшіліктері:

- сөздіктің көлемі артқан сайын, алгоритм жұмысы қиындайды;
- жалғыз символдарды кодтау тиімсіз болады.

3.4.2 LZ78 (Лемпель-Зив) алгоритмі

1978 жылы LZ78 алгоритмі пайда болды. LZ78 «жылжымалы» терезелерді қолданбайды, оның сөздігі қарастырылып өткен жолдардан құралады. Ол қарастырылған мәтіндегі кезіккен тізбектерді сөздікке енгізеді. Алгоритм басталғанда бұл сөздік бір бос жолдан тұрады. Жол сөздікке толығымен жазылмайынша, алгоритм ақпараттың символдарын есептей береді. Кодтау үшін сөздіктен тізбекті қарайды, егер тізбек сөздіктен табылса, нәтижеге

осы жолдың коды беріледі. Егер тізбек сөздіктегі жазылғанның толықтырылған түрі болса, онда тізбек сөздікке енгізіледі.

Алгоритм хабардың символдарын сөздікпен сәйкес келетін жол табылғанша жинайды да оларды қосады. Табылған жол сөздіктегі жолдан кем дегенде бір символға ерекшеленсе, алгоритм сөздіктегі жолдың индексінен тұратын кодты өндейді. Егер сөздік толған болса, ондағы барлық салыстырмалы жолдар жойылады.

Сөздіктегі жолдардың көлемі негізгі код болып табылады, себебі LZ78 әдісі бойынша кодтағанда әрбір код сөздіктегі жолдың нөмірін қамтиды. Бұл кодтардың тұрақты ұзындығы екілік логарифмделген дөңгелектелген сөздік +8 көлеміне тең болады (ASCII кеңейтілуімен қолданылатын байт-кодтағы биттердің көлемі).

Мысалы, сөздікте А тізбегі сақталса, ал алгоритм АВ тізбегін кезіктірсе, онда алдымен сөздіктегі А тізбегінің коды шығарылады, содан кейін АВ сөздікке енгізіледі. Егер кейін ABC кезіксе, онда нәтижеге АВ үшін жазылған код шығарылады, сөздікке жаңа тізбекті ABC енгізіледі, оның ұзындығы 1 байтка ұзын болады.

LZ78 алгоритмімен кодтау. Төмендегі 3.8-кестеде КРАСНАЯ КРАСКА мәтінін LZ78 алгоритмімен кодтаудың мысалы көрсетілген (мысал [3] әдебиеттен алынды). Сөздіктің ұзындығы 16 фразаға тең. Мұндай сөздіктегі кез-келген жолдың сілтемесі 0 ден 15 дейінгі сандар, оны кодтау үшін 4 бит жеткілікті. Соңғы мысалдағы алынған кодтың ұзындығы $10*(4+8)=120$ бит.

Кесте 3.8 – LZ77 алгоритмімен кодтау

| Кіру фразасы (Сөздіктің ұзындығы – 16) | LZ78 коды ұзындығы $-10*(4+8)=120$ бит. | Сөздіктің орны |
|---|--|----------------|
| « » | | 0 |
| « К » | 0, К | 1 |
| « Р » | 0, Р | 2 |
| « А » | 0, А | 3 |
| « С » | 0, С | 4 |
| « Н » | 0, Н | 5 |
| « АЯ » | 3, Я | 6 |
| « » | 0, '' | 7 |
| « КР » | 1, Р | 8 |
| « АС » | 3, С | 9 |
| « КА » | 1, А | 10 |

LZ78 кодын декодтау. LZ78 алгоритмі бойынша сығылған хабардың кодтары – <0 ,«К»> <0 ,«Р»> <0 ,«А»> <0 ,«С»> <0 ,«Н»> <0 ,«Я»> <0 ,« »> <1 ,«Р»> <3 ,«С»> <1 ,«А»> берілсін.

Сөздіктің ұзындығы – 16 жол. Төмендегі 3.9-кестеде LZ78 алгоритмімен кодталған хабарды декодтаудың мысалы көрсетілген (мысал [3] әдебиеттен алынды).

Кесте 3.9 – LZ78 алгоритмімен сығылған хабарды декодтау

| Сығылған кодтары | хабар | Декодталуы | Сөздіктің орны |
|------------------|-------|------------|----------------|
| <0 ,«К»> | | « » | 0 |
| <0 ,«Р»> | | «К» | 1 |
| <0 ,«А»> | | «Р» | 2 |
| <0 ,«С»> | | «А» | 3 |
| <0 ,«Н»> | | «С» | 4 |
| <3 ,«Я»> | | «Н» | 5 |
| <0 ,« »> | | «АЯ» | 6 |
| <1 ,«Р»> | | « » | 7 |
| <3 ,«С»> | | «КР » | 8 |
| <1 ,«А»> | | «АС» | 9 |
| | | «КА» | 10 |

3.4.3 LZSS (Лемпель-Зив-Сторер-Шиманский) алгоритмі

1982 жылы Сторер мен Шиманский LZ77 негізінде LZSS алгоритмін жасады. Бұл танымал код басқа сығу алгоритмдерімен бірге қолданылады, мысалы: PKZIP V.1.0 архиваторында LZSS пен Шеннон-Фано алгоритмдерінің комбинациясы, ал ARJ архиваторында – LZSS пен Хаффман алгоритмі қолданылған.

LZSS алгоритмімен кодтау. LZSS коды бір битті префикстен басталады, ол кодты кодталмаған символдан ажыратады.

LZSS коды жұптан, яғни LZ77 алгоритміндегідей ығысу және ұзындықтан тұрады. Терезе табылған фразаның ұзындығына жылжиды немесе буфердегі сөздіктен табылмаса 1 символға солға жылжиды. Фраза ұзындығы әрқашан нольден үлкен болады.

Декодерді сығылған хабардың декодтауы біткен соң тоқтату үшін кодер сығылған файл соңына арнайы символ орналастырады.

Кодтау мысалы. Төмендегі 3.10-кестеде КРАСНАЯ КРАСКА мәтінін LZSS алгоритмімен кодтаудың мысалы көрсетілген (мысал [3] әдебиеттен алынды). Бұл мысалда мәтіннің бір символы 8 битпен кодталады, сондықтан «0» префиксінен кейін келесі 8 бит оқылады, ал «1» префиксінен кейін *(ығысу, ұзындық) жұбы* сақталады.

Ығысу мен ұзындық сөздіктің ұзындығынан аспайтын болғандықтан, *(ығысу, ұзындық) жұбының* өлшемі сөздіктің ұзындығын сақтауға қажетті биттердің екі еселенуіне тең болады. Сөздіктің ұзындығы 8 бит болғандықтан, «1» префиксінен кейін декодер 6 бит оқиды.

Кесте 3.10 – LZSS алгоритмімен кодтау

| Сөздік (ұзындығы 8 бит) | Буфер (ұзындығы 5 бит) | LZSS коды | Код ұзындығы |
|----------------------------|---------------------------|-----------|-----------------|
| «.....» | « КРАСН » | 0, К | 9 |
| «.....К» | « РАСНА » | 0, Р | 9 |
| «.....КР» | « АСНАЯ » | 0, А | 9 |
| «.....КРА» | « СНАЯ » | 0, С | 9 |
| «....КРАС» | « НАЯ К » | 0, Н | 9 |
| «...КРАСН» | « АЯ КР » | 1, 5, 1 | 7 |
| «. .КРАСНА» | « Я КРА » | 0, Я | 9 |
| «. .КРАСНАЯ» | « КРАС » | 0, ' ' | 9 |
| « КРАСНАЯ» | « КРАСК » | 1, 0, 4 | 7 |
| « НАЯ КРАС» | « КА...» | 1, 4, 1 | 7 |
| « АЯ КРАСК» | « А....» | 1, 0, 1 | 7 |

Кодтың ұзындығы $7*9 + 4*7=91$ бит.

LZSS кодын декодтау. LZSS алгоритмі асимметриялы. Сығу алгоритмі күрделі болғанымен, LZSS декодері қарапайым.

Декодер 1 бит сығылған хабарды оқып, оның символ немесе жұбы ма екенін анықтайды.

Егер символ болса, келесі 8 бит декодталған символ ретінде жылжымалы терезеге жазылады. Ал *(ығысу, ұзындық)* түріндегі жұп болса (яғни файл соңы болмаса), онда сөздіктегі сәйкес символдар терезеге жазылады да, декодталып беріледі.

Декодтау мысалы. Сөздіктің ұзындығы – 8 символ (бит). LZSS алгоритмімен сығылған хабардың кодтары – 0«К» 0«Р» 0«А» 0«С» 0«Н» 1<5,1> 0«Я» 0« » 1<0,4> 1<4,1> 1<0,1> берілген.

Төмендегі 3.11-кестеде LZSS алгоритмімен кодталған осы хабарды декодтаудың мысалы көрсетілген (мысал [3] әдебиеттен алынды).

Кесте 3.11 –LZSS алгоритмімен сығылған хабарды декодтау

| Сығылған кодтары | хабар | Декодталуы | Сөздік |
|------------------|-------|------------|------------|
| 0«Р» | | «К» | «.....К» |
| 0«А» | | «Р» | «.....КР» |
| 0«С» | | «А» | «.....КРА» |
| 0«Н» | | «С» | «...КРАС» |
| 1<5,1> | | «Н» | «...КРАСН» |
| 0«Я» | | «А» | «.КРАСНА» |
| 0« » | | «Я» | «.КРАСНАЯ» |
| 1<0,4> | | « » | «КРАСНАЯ » |
| 1<4,1> | | «КРАС » | «НАЯ КРАС» |
| 1<0,1> | | «К» | «АЯ КРАСК» |
| | | «А» | «Я КРАСКА» |

LZ77 пен LZSS алгоритмдерінің кемшіліктері:

– Егер екі жолдың арақашықтығы сөздіктін ұзындығынан үлкен болса, оларды кодтау мүмкін емес.

– Кодталатын жолдың ұзындығы буфердің көлемімен шектеледі.

Егер сөздік пен буфер көлемін механикалық түрде ұлғайтса, онда кодтаудың тиімділігі соғұрлым төмендейді, себебі бұл көрсеткіштер артқан сайын ығысатын кодтың ұзындығы да артады. Сонымен қатар, алгоритмнің (яғни кодердің) жұмысының уақыты да артады.

1978 жылы Лемпель мен Зивтің ұсынған LZ78 алгоритмінде жоғарыда аталған кемшіліктер болмады.

3.4.4 LZW (Лемпель-Зив-Уэлч) алгоритмі

1984 жылы Уэлч (Welch) LZ78 алгоритмін модификациялау арқылы LZW алгоритмін құрды. Деректерді сығудың бұл алгоритмі TIFF, PDF, GIF, PostScript және басқа да форматтағы файлдарда, ZIP, ARJ, LHA сияқты танымал сығу программаларында

колданылады. Графиктік аппараттарды сығу дәрежесі жоғары. LZW патенттелген алгоритм болып табылады. Оны аппараттық деңгейде лицензиясыз қолдануға болмайды.

LZ78 әдісіндегідей LZW алгоритмінде алынатын кодтың көлемі үшін негізгі код жолдардағы сөздіктің көлемі болып табылады: LZW алгоритмі өзінің жұмысында 4096 элементтен тұратын сөздікті қолданады.

LZW коды – сөздік көлемінің екілік логарифміне тең болатын тұрақты ұзындығына тең.

LZW алгоритмі келесі қадамдарды жүзеге асырудан тұрады:

1-қадам. Барлық мүмкін болатын бір символдық жолдарды сөздікке жазу (әдетте ASCII кеңейтілуінде қолданылатын 256 символмен инициализацияланады). Кірістік фразаны (*w*, яғни жолды) хабардың бірінші символымен инициализациялау.

2-қадам. Кодталатын хабардан кезекті *K* символды санау.

3-қадам. Егер **ХАБАРДЫҢ_СОҢЫ** болса, онда *w* үшін код беру.

4-қадам. Егер сөздікте *wK* фразасы (жолы) бар болса, онда

- кірістік фразаға (кодталатын жолға) *wK* мәнін меншіктеу;
- 2-ші қадамға өту;

әйтпесе:

- *w* кодын беру;
- *wK* фразасын сөздікке қосу;
- кірістік фразаға (кодталатын жолға) *K* мәнін меншіктеу;
- 2-ші қадамға өту.

5-қадам. Соңы.

LZW алгоритмімен кодтау. LZW алгоритмі арқылы **КРАСНАЯ КРАСКА** мәтінін (кірістік фразаны) кодтаудың мысалы 3.12-кестеде көрсетілген (мысал [3] әдебиеттен алынды). Бұл мысалда алынған кодтың ұзындығы $12 \cdot 9 = 108$ битке тең. Сөздік толып кеткен жағдайда толған сөздікке жаңа жол енгізу үшін сөздіктен ең сирек қолданылатын жолдар немесе бірлік символдан басқа барлық жолдар жойылады.

Сөздіктегі 0 мен 255 аралығындағы нөмірлі элементтер бөлек символдардан құралады. Сөздіктің бұл элементтері тұрақты, яғни алгоритмнің жұмыс барысында өзгермейді. Сөздіктің қалған 256-дан бастап 4095-ші нөмірге дейінгі элементтері бастапқыда бос

сілтемеге ие болады. Ал жұмыс барысында келесі ереже бойынша құралған фразаларға көрсеткіш бола алады: ағымдағы фразаның соңына ағымдағы символдың қосылуымен жаңа фраза жасалынады. Сөздік толған кезде LZW алгоритмі өзінің жұмысын басынан бастаған сияқты 256-4095 нөмірлі элементтеріне нөлдік сілтемелерді жазады.

Кесте 3.12– LZW алгоритмі арқылы кодтау

| Кодталатын жол wK (сөздікке кіру фразасы) | w үшін LZW коды Код ұзындығы: 12*9=108 бит | Сөздіктегі орын |
|--|---|--------------------|
| ASCII+ | | 0-255 |
| «KP» | 0 «K» | 256 |
| «PA» | 0 «P» | 257 |
| «AC» | 0 «A» | 258 |
| «CH» | 0 «C» | 259 |
| «HA» | 0 «H» | 260 |
| «АЯ» | 0 «А» | 261 |
| «Я » | 0 «Я» | 262 |
| « K» | 0 « » | 263 |
| «KPA» | <256> | 264 |
| «ACK» | <258> | 265 |
| «KA» | 0 «K» | 266 |
| «A» | 0 «A» | |

Декодтау. Сөздіктің ұзындығы – 8 байт (символ). LZW алгоритмімен сығылған хабардың кодтары – 0«K», 0«P», 0«A», 0«C», 0«H», 0«A», 0«Я», 0« », <256>, <258> 0«K», 0«A» берілсін. Төмендегі 3.13-кестеде LZSS алгоритмімен кодталған хабарды декодтаудың мысалы көрсетілген (мысал [3] әдебиеттен алынды).

Кесте 3.13 – LZW алгоритмімен кодталған хабарды декодтау

| Сығылған хабар | Декодталуы | Сөздік | Сөздіктің орны |
|----------------|------------|--------|----------------|
| | «K» | ASCII+ | 0-255 |
| 0 «K» | «P» | «KP» | 256 |
| 0 «P» | «A» | «PA» | 257 |
| 0 «A» | «C» | «AC» | 258 |
| 0 «C» | «H» | «CH» | 259 |
| 0 «H» | «A» | «HA» | 260 |

| | | | |
|-------|-------|-------|-----|
| 0 «А» | «Я» | «АЯ» | 261 |
| 0 «Я» | « » | «Я » | 262 |
| 0 « » | «КР » | « К» | 263 |
| <256> | «АС» | «КРА» | 264 |
| <258> | «К» | «АСК» | 265 |
| 0 «К» | «А» | «КА» | 266 |

LZW алгоритмінің артықшылықтары:

– Символдардың немесе кодтардың хабарда кездесу ықтималдықтарын есептеуді қажет етпейді.

– Декомпрессия (сығылған хабардан бастапқы хабарды алу) үшін, жолдардың кестесін ашу үшін файлға жазудың қажеті жоқ. Бұл алгоритм арқылы жолдар кестесін тек кодтар ағынын пайдаланып қайта қалпына келтіруге болады.

– Сығудың (компрессия) бұл түрі бастапқы графикалық файлға өзгеріс енгізбейді, кез келген типтегі растрлық деректерді сығуға жарамды.

LZW алгоритмінің кемшілігі: алгоритм кірістік деректерге талдау жасамағандықтан, тиімді емес.

БӨЛІМ БОЙЫНША БАҚЫЛАУ СҰРАҚТАРЫ:

1. Деректерді сығу деген не? Сығу не үшін, қайда қолданылады?
2. Деректерді сығудың қандай екі негізгі типі бар? Сипаттама беріңіз.
3. Қандай жағдайда ақпаратты сығуға болады?
4. Артықтықтың қандай түрлері бар? Сипаттама беріңіз.
5. Ақпаратты сығу үшін қандай арнаулы сығушы программалар (архиваторлар) қолданылады? Оларға қандай сығылған файлдардың кеңейтілімдері және сығу әдістері сәйкес келеді?
6. Қандай сығу түрлері бар? Сипаттамаларын беріңіз.
7. Мәтіндік ақпаратты сығудың қандай тәсілдері бар?
8. Деректерді сығудың қандай әдістері мен құралдары бар?
9. Символдар статистикасын қолдануына байланысты деректерді сығу алгоритмдері қандай 3 топқа бөлінеді? Сипаттама беріңіз.
10. Сығу дәрежесі деген не? Қалай аныкталады?
11. Арифметикалық кодтаудың ерекшелігі неде? Қолданысын, кодтау және декодтау алгоритмдерін сипаттаңыз.
12. Адаптивті арифметикалық кодтау мен кодты декодтаудың алгоритмдерін сипаттаңыз. Арифметикалық кодтаудан оның айырмашылығы неде?
13. Лемпель-Зив алгоритмдерінің қандай нұсқалары бар?
14. Лемпель-Зив алгоритмдерінің қолданысын, кодтау және декодтау алгоритмдерін сипаттаңыз.
15. Лемпель-Зив алгоритмдерінің қандай артықшылықтары мен кемшіліктері бар?

ӨЗДІК ЖҰМЫС ТАПСЫРМАЛАРЫ

1. Әріптеп сығуға, сөздер мен сөз тіркестерін сығуға, мәтінді сығу және жималауға, сандар массивін сығуға мысалдар келтіріңіз.

2. AABCDAAACCCDBB хабарын арифметикалық кодпен кодтаңыз.

3. "КАВАВАВАВАВZ" хабарын адаптивті арифметикалық кодпен кодтаңыз.

4. Арифметикалық кодтау арқылы жасалған 0.41147 кодын декодтаңыз.

5. "AABCDAAACCCDBB", "КАВАВАВАВАВZ", "КІТАПХАНАҒА ЖАҢА КІТАПТАР ТҮСТІ" және "БАЛАЛАР БАЛ АЛДЫ" хабарларын келесі сығу алгоритмдерін қолданып кодтаңыз және кодталған хабардың ұзындығын битпен есептеңіз:

a) LZ77 (сөздік - 12 байт, буфер - 4 байт);

b) LZ78 (сөздік - 16 фраза);

c) LZSS (сөздік - 12 байт, буфер - 4 байт);

d) LZW (сөздік - ASCII+ және 16 фраза).

6. Төмендегі Лимпель-Зив алгоритмдерімен кодталған хабарларды декодтаңыз және әрбір сығылған хабар кодының ұзындығын битпен есептеңіз.

a) LZ77 (сөздік - 12 байт, буфер - 4 байт) алгоритмімен сығылған хабар коды $\langle 0,0,'A' \rangle \langle 0,0,'F' \rangle \langle 0,0,'X' \rangle \langle 9,2,'F' \rangle \langle 8,1,'F' \rangle \langle 6,2,'X' \rangle \langle 4,3,'A' \rangle$

b) LZSS (сөздік - 12 байт, буфер - 4 байт) алгоритмімен сығылған хабар коды $- 0'A' 0'F' 0'X' 1\langle 9,2 \rangle 1\langle 8,2 \rangle 1\langle 6,3 \rangle 1\langle 4,4 \rangle 1\langle 9,1 \rangle$

c) LZ78 (сөздік - 16 фраза) алгоритмімен сығылған хабар коды $\langle 0,'A' \rangle \langle 0,'F' \rangle \langle 0,'X' \rangle \langle 1,'F' \rangle \langle 2,'X' \rangle \langle 5,'A' \rangle \langle 3,'A' \rangle \langle 2,'F' \rangle \langle 0,'A' \rangle$

d) LZW (сөздік - ASCII+ және 16 фраза) алгоритмімен сығылған хабар коды $- 0'A' 0'F' 0'X' \langle 256 \rangle \langle 257 \rangle \langle 257 \rangle 0'A' \langle 258 \rangle 0'F' 0'F' 0'A'$

3.5 Графикалық ақпаратты сығу әдістері

3.5.1 Кескіндерді сығу

Графикалық ақпараттарды, яғни кескіндер мен беінені (video) сығуда ескерілетін олардың келесі ерекшеліктері бар:

1) мәтінмен салыстырғанда олар жадта көп орын алады, сондықтан да оларды сақтау, тасымалдау үшін сығу өзесті;

2) адам көзі суретті қараған кезде контурға, түстерге мән береді де, кескіндегі аздаған өзгерістерді байқамайды. Бұл адам көзінің кескінді қабылдау ерекшелігіне негізделген, декомпрессияланған кескін түпнұсқамен бірдей болмайтындай етіп кескінді сығатын тиімді алгоритмдердің жасалуына әкелді;

3) мәтінге қарағанда кескіннің 2 өлшемде артықтығы болады, яғни көрші тұрған нүктелердің тігінен және көлденеңінен түстері бір-біріне жақын болады. Сондықтан графиканы сығу алгоритмдерінде кескіннің құрылымы да ескеріледі.

Кескіндердің шартты түрде жіктелуі:

- *Түстердің саны шағын (4-16 түс) кескіндер және бір түспен үлкен аудандары боялған кескіндер.* Түстердің баяу ауысуы болмайды. Мысалы: іскер графика – диаграмма, гистограмма, график және т.б.

- *Компьютер арқылы салынған, түстердің баяу ауысуы болатын кескіндер.* Мысалы: презентациялардағы графика, САПР-дағы эскиздік моделдер, Гуро әдісімен салынған суреттер.

- *Фотореалистік кескіндер* (сканерленген фотосуреттер).

- *Іскер графика қосылған фотореалистік кескіндер* (жарнама).

Бұл кескіндердің әрқайсысының жеке тобына арналған тиімді сығу алгоритмдері бар, сондықтан да графиклық сығу алгоритмдерін салыстырғанда кескіндердің қандай класына арналғанын ескеру маңызды.

Сығу сапасының деңгейлерін (немесе оны *сығу дәрежелері* деп атайды) қарастырайық.

Сапасын жоғалтпай сығу – мұндай сығуда сығу процесі кезінде деректер жоғалмайтын жағдайда ғана кескіндерді (сурет, image) сығу жүзеге асады. Нәтижесінде декомпрессиядан соң алынған кескін сығуға дейінгі түпнұсқамен әрбір бит бойынша бірдей болады. Сығудың мұндай түрінің мысалына статикалық

(қозғалыссыз) кескінге арналған GIF форматы мен бейнеге арналған GIF89 жатады.

Сапасын жоғалтып сығу – мұндай сығуда сығу процесі кезінде деректер жоғалатындықтан кескіннің сапасы төмендейді. Бұған адамның суретті қабылдау ерекшелігіне байланысты тек адам көзі түпнұсқадан нәтижелік суретті ажырата алатын сығулар жатады. Кескіннің түпнұсқасы мен қандай да бір компрессор арқылы оның сығылған нәтижесі әрбір биті бойынша салыстырғанда сәйкес болмаса да, олардың арасындағы айырмашылық бір қараған кезде байқалмауы да мүмкін. Мұның мысалына статикалық кескінді сығуға арналған JPEG алгоритмі мен бейнені сығуға арналған M-JPEG жатады.

Қабылдау тұрғысынан сапасын жоғалтпай сығу – мұндай сығуда шартты түрде сапасын жоғалтатын сығу бола тұра, адамның қабылдауы тұрғысынан сапасын жоғалтпай сығу сияқты болып көрінуі мүмкін. Осындай формальды түрде сапасын жоғалтып сығатын технологиялардың көпшілігінде сапаның қабылданатын жағын 0-ден 100-ге дейінгі аралықта сипаттайтын *сығу сапасының факторы* болады. Сығу сапасының факторы 100-ге тең болғанда сығылған ақпараттың қабылданған сапасы түпнұсқадан ажырата алмайтындай болады.

Табиғи түрде сапасын жоғалтып сығу. JPEG пен MPEG және сапасын жоғалтып сығатын басқа да сығу технологиялары кейде бейнеақпаратты қабылдау тұрғысынан сығудың шегінен асырып сапасын жоғалтпай сығады. Соған қарамастан, сығылған бейне мен статикалық кескіндер адамның оны дәлме-дәл қабылдауына толық жарамды болады, яғни бұл жағдайда айналасындағы кейбір ұсақ детальдары жоғалатын кескіннің табиғи деградациясы байқалады. Осындай сығуға ұқсас нәтиже табиғи жағдайларда (мысалы, жаңбыр немесе тұман кезінде түсірілген бейнеде) кездесуі мүмкін, мұндайда кескін көрінгенімен, оның бөлшектенуі (детализациясы) азаяды.

Табиғи емес түрде сапасын жоғалтып сығу. Бұл сығу түрінде сығу сапасы төмен болғандықтан, кескін айтарлықтай бұрмаланады және кескінге жасанды түрде (түпнұсқасында болмайтын) оның айналасының бөлшектері енгізіледі. Мұндай сығу мысалына ЖДКТ технологиясын қолданатын кейбір қатты сығылған MPEG пен басқа компрессорлардағы «блоктылық» жатады. Табиғи еместілігі

алдымен адамның қабылдау тұрғысынан кескіндегі ең маңызды сипаттаманың – *контурлардың* жоғалуынан байқалады. Осы контурлар адамның қабылдаушы аппаратына көріп тұрған затты немесе шолу нысанын көзбен дұрыс идентификациялауға көмектеседі.

Графикалық ақпаратты сығушы алгоритмдерді қолданатын жүйелердің шартты түрде 3 тобы бар:

1. *Компрессия мен декомпрессия уақытына жоғары талап қоятын жүйелер*, олардың кейбіреуі кескіннің кішірейтілген көшірмесін қарауды, суретті кескіндер базасынан іздеуді қажет етеді. Мысалы: баспа жүйелері әр текті иллюстрацияларды, фотореалистік суреттер мен іскер графика көп қолданылатындықтан оларды сақтау проблемасы туындайды, сондықтан әмбебап сығу алгоритмдері қолданылады.

2. *Компрессия дәрежесіне және декомпрессия уақытына жоғары талап қоятын жүйелер*. Сығу уақыты маңызды емес. Мысалы: CD-ROM-дағы сөздіктер мен энциклопедиялар (бейне, статикалық кескін түрінде);

3. *Компрессия дәрежесіне жоғары талап қоятын жүйелер*. Мысалы: геоақпараттық жүйелердегі жер бедерінің аэрофототүсірімдерінде суреттердің өлшемі аса үлкен болғандықтан, тек бөлігін ғана көру, масштабтау қажет болады. WWW-дегі түрлі-түсті иллюстрациялар, желі арқылы серверден графикалық ақпарат алу, экранда рентген мен ультрадыбыс суреттерін талдау, электронды түрдегі пациенттің ауруының тарихы, түрлі фотодеректер картотекасынан іздеу, адамның әртүрлі қырынан түсірілген фотолардан ұқсастығын анықтау сияқты және т.б. атқаратын жүйелер.

Жоғарыда аталған жүйелердегі графикалық ақпараттардың қолданылу ерекшеліктері сығу алгоритмдеріне өз шектеулерін қояды.

Сондықтан да сығу алгоритмдерін салыстырғанда қандай жүйелерде, қандай кескіндерді сығу үшін қолданылатынын білу ғана емес, сондай-ақ сығудың жоғары дәрежесі болуы, кескіннің жоғары сапасы болуы, сығудың жоғары жылдамдығы болуы, декомпрессияның жоғары жылдамдығы болуы, кескіндердің өзіндік ерекшелігін ескеру, масштабталу болуы, редакторлеу мүмкіндігінің болуы, желідегі суретті жүктемес бұрын суретті

алдын-ала көрсету, алгоритмнің аппараттық жүзеге асу құнының арзандығы, программалық жүзеге асуының тиімділігі, қателерге төзімділігі (файлдағы кескіннің бүлінуі немесе фрагментінің жоғалуы болғанда) сияқты талаптар ескеріледі.

Кескінді сығу алгоритмдері *шығынмен сығу және шығынсыз сығу* болып 2 топқа бөлінеді.

Кескінді шығынсыз сығу алгоритмдеріне RLE, LZW алгоритмдері, JBIG Lossless JPEG жатады.

Кескінді шығынмен сығу алгоритмдеріне JPEG, JPEG-2000 алгоритмдері, фрактальды алгоритм, рекурсиялық (толқындық) алгоритм жатады.

Ескерту. Толығырақ танысу үшін қараңыз: Ватолин Д., Ратушняк А., Смирнов М., Юкин В. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео. - М.: ДИАЛОГ-МИФИ, 2002. - 384 с.

3.5.2 Бейне ақпаратты сығу

Цифрлық бейнені (видео) сығудың көптеген технологиялары бар. Кейбір компрессорлар сығудың тек бір технологиясын ғана емес, олардың жиынтықтарын қолданады. Мысалы, Cinepak, Indeo 3.2 компрессорлары векторлық квантизацияны қолданса, ал MPEG-1, MPEG-2, MPEG-4, H.261 және H.263 халықаралық стандарттары жылдам *дискретті косинустық түрлендірудің* (DCT – Discrete-Cosine Transform,) біріктірілген технологиясы мен қозғалыс компенсациясын қолданады. Кейбір қазіргі заманғы алгоритмдерде *дискретті вейвлетті түрлендіру* (DWT – Discrete Wavelet Transform) технологиясы қолданылған. Басқа технологиялар кескінді фрактальды қысқартуға (Fractal Image Compression) негізделген.

Кеңінен қолданылып жүрген барлық бейнекомпрессорлар сапасын жоғалтып сығу технологияларын қолданады. Сығудың қажетті жоғары коэффициенттерінде олардың барлығы жасанды түрде сапасынан айырып сыға бастайды.

Сондықтан, цифрлық бейнені сығу үшін компрессорды таңдаған кезде сапасы табиғи түрде жоғалатындай етіп сығуға тырысу керек.

Бейнені сығу стандарттарына Motion-JPEG, MPEG-1, H.261, H.263, MPEG-2, MPEG-4 жатады.

Ескерту. Толығырақ білу үшін мына әдебиетті қараңыз: Ватолин Д., Ратушняк А., Смирнов М., Юкин В. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео. - М.: ДИАЛОГ-МИФИ, 2002. - 384 с. және әзірлеушілердің сайттарын қараңыз.

Кодектер (компрессорлар мен декомпрессорлар сөздерінен қысқартылған) – деректерді сығу және кері қалпына келтіру технологиясы. Қандай да бір бейне файлды іске қосу үшін компьютерде (мобилді құрылғыда) осындай форматты қолдайтын кодек орнатылуы тиіс. Мысалы, ISO MPEG-4 стандартын кроссплатформалы (Windows, Linux and Mac үшін) Xvid кодек жүзеге асырады, оның бейнені сығу дәрежесі – 200:1.

Ескерту. Кодектердің соңғы версияларының мүмкіндіктері, оларды жүктеу туралы http://codecs.necromancers.ru/index_r.html сайты және кодектерді әзірлеушілердің сайттарын қараңыз.

3.5.3 Графикалық ақпаратты сығу алгоритмдері, технологиялары

Run Length Encoding технологиясы (RLE, қайталауларды кодтау немесе сериялардың ұзындықтарын кодтау). RLE–қайталанатын символдарды (серияларды) бір символмен және олардың санымен алмастыратын сығу алгоритмі. Серияларды құрайтын бірдей символдардан тұратын жол қайталанатын символ мен қайталану санынан тұратын жолмен кодталады. Мұндай қайталанулар (сериялар) иконкалар мен графикалық суреттерде өте көп кездесетіндіктен, қарапайым суреттерді сығуда аса тиімді.

RLE алгоритмі PackBits, PCX, ILBM форматтарында, сондай-ақ ұзын байттар тізбегінен тұратын алдын ала дельта-кодтаудан өткен дыбыстық деректерді сығуда қолданады.

RLE технологиясын қолданатын компрессорларға Microsoft RLE (MRLE) жатады. RLE алгоритмі MPEG-1-2-3-4, H.261, H.263 және JPEG-те қолданылатын БДКП-де коэффициенттерді кодтау үшін де қолданылады.

Артықшылықтары мен кемшіліктері:

– тек бірыңғай 8-биттік кескіндермен жұмыс жасайды;

– толық түрлі-түсті бейнені қысқартуға жарамсыз.

RLE кескіндегі қайталанатын элементтерді немесе бірдей түсті элементтерді бір кодтық сөзбен кодтайды. Мысалы, кескіндегі 11 11 11 11 11 элементтерінің тізбегі алты 11 түрінде кодталады. RLE контурларының қайталануы мен жеке элементтердің түстерінің қайталануы кездесетін кескіндерді жақсы сығады. Ал, толық түрлі түсті кескіндерде түстердің қайталанулары сирек болатындықтан, толық түрлі түсті бейнені RLE технологиясын қолданып сығу тиімсіз.

Векторлық квантизация технологиясы (Vector Quantization, VQ). VQ технологиясын қолданатын компрессорларға Indeo 3.2 және Сіперак жатады, бұлар YUV (RGB емес) түстер схемасын қолданады.

Артықшылықтары мен кемшіліктері:

– кодтау процесі аса қиын және арнайы қосымша жабдықтарсыз жүзеге аспайды;

– декодтау процесі өте жылдам жүреді;

– сығудың жоғары коэффициенттерінде блоктық бұрмаланулар болады;

– DCT (жылдам дискретті косинустық түрлендіру), DWT (дискретті вейвлетті түрлендіру) алгоритмдерін қолданатын технологиялар бұдан да жоғары деңгейде сыға алады.

Векторлық квантизация кескінді блоктарға бөлуге негізделген (мысалы: Indeo мен Сіперак компрессорларында YUV түстер схемасында 4x4 өлшемді пикселдік блоктар). Кейбір блоктар бір біріне ұқсас болып шығуы мүмкін. Бұл жағдайда компрессор ұқсас блоктардың тобын идентификациялайды да оларды бір ортақ блокпен ауыстырады және ең қысқа кодталған сөздерден тұратын осындай ортақ блоктардың екілік кестесі (картасы) құрылады. Осы кестені қолдана отырып VQ-декодер сығылған ақпаратты декодтау кезінде ортақ блоктардан кескінді блоктап жинап алады. Кодтаудың бұл түрінде кескіннің сапасы жоғалып сығылады, себебі блоктардың ұқсастығы шартты түрде ғана, мұнда нақты блоктардың орнына олардың біріктірілген ортақ кескін түріндегі аппроксимация қолданылады. Кодтау процесі күрделі, ұзақ уақыт жұмсауды қажет етеді, өйткені кодер кескіннің әрбір блогының қандай да бір ортақ блокқа сәйкес келуін анықтап отырады. Ал, декодтау бұл жағдайда ортақ блоктардан берілген карта бойынша кескінді құруға

негізделеді және аппараттық және уақыттық ресурстарды аса қажет етпейді. Кесте немесе картаны *кодтық кітап*, ал оның ішіндегі екілік кодтарды *кодтық сөздер* деп те атайды. VQ алгоритмін қолдана отырып сығудың жоғары деңгейіне ортақ блоктардың топтарының санын азайту арқылы қол жеткізеді, нәтижесіне кодтық кітап көлемі де кішірейеді. Ал кодтық кітап көлемі кішірейген сайын бейненің қалпына келу сапасы төмендейді. Нәтижесінде кескінде жасанды «блоктылық» пайда болады. (Vector Quantization and Signal Compression, A. Gersho and R. Gray, Boston, MA : Kluwer, 1992).

Төмендегі 3.14-кестеде 4x4 өлшемді үш блоктың салыстырылуы көрсетілген. Адам көзіне бір қарағанда бұл блоктар бірдей сияқты көрінеді, сондықтан 2-ші, 3-ші блоктарды 1-ші блокпен алмастыруға болады.

Кесте 3.14 – Үш 4x4 блоктардың салыстырылуы

| (1-блок) | (2-блок) | (3-блок) |
|-----------------|-----------------|-----------------|
| 128 128 128 128 | 128 127 128 128 | 128 127 126 128 |
| 128 128 128 128 | 128 128 128 128 | 128 128 128 128 |
| 128 128 128 128 | 128 128 127 128 | 127 128 128 128 |
| 128 128 128 128 | 128 128 128 128 | 128 128 128 128 |

Нәтижесінде кодтық кітап келесі түрде болады:

128 128 128 128

128 128 128 128

128 128 128 128

128 128 128 128

VQ технологиясының маңызды ерекшелігі: бейнені сығу кезінде бір кодтық кітапты кескіннің бірнеше кадрлары үшін қолдануға болады.

Дискреттік косинустық түрлендіру (DCT, Discrete-Cosine Transform) технологиясы. ДКТ қолданатын компрессорларға Motion JPEG; Editable MPEG; MPEG-1; MPEG-2; MPEG-4 жатады.

Артықшылықтары мен кемшіліктері:

- жоғары компрессия кезіндегі «блоктылық» болады;
- кескіннің үшкір бұрыштарының ұштары доғалданады, кескіннің үшкір шеттері кездейсоқ түрде жоғалып та кетеді;

– кодтауға көп уақыт жұмсалады, тек соңғы кездерде ғана кодтау процесін аппаратты түрде емес, программалық түрде жүзеге асыру мүмкін болды.

Кескіндерді сығуда ДСТ технологиясы кеңінен қолданылады, атап айтқанда: бейнеконференцияларда қолданылатын H.263 стандартында, MPEG цифрлық бейнестандартында (MPEG–1, MPEG–2 и MPEG–4), JPEG статикалық суреттерді сығу стандартында. Бұл стандарттарда өлшемдері 8×8 пиксель болатын кескін блоктарына қолданылатын екі өлшемді ДСТ қолданылады. ДСТ 64 коэффициентті (8×8=64) есептеп, содан кейін квантизациялап, нақты түрде сығуды қамтамасыз етеді. Көптеген кескіндерде ДСТ коэффициенттерінің көпшілігі өзінің мәні аз болғандықтан квантизациядан соң нөлге тең болады. ДСТ -нің осы қасиеті ДСТ қолданатын көптеген сығу алгоритмдерінің негізінде жатыр.

Сонымен қатар, ДСТ -нің үлкен коэффициенттерімен берілетін жоғары жиілікті кескіннің компоненттеріне адам көзінің сезімталдығы аз болып табылады. Ал, коэффициенттердің осы үлкен мәндеріне әдетте квантизацияның үлкен факторы қолданылады. Негізінде JPEG алгоритмінде қолданылатын ДСТ -нің 64 коэффициенттерінің әрқайсысына арналған 64 квантизация факторларының матрицасында ДСТ коэффициенттері үшін квантизация факторлары үлкен болады, сәйкесінше, жоғары жиілікке ие. Квантизациядан соң коэффициенттерді RLE алгоритмі өңдейді. Ары қарай жиі кездесетін комбинациялар үшін қысқа кодтық сөздер, ал сирек кездесетіндері үшін ұзын кодтық сөздер қолданылады. Мұнда ықтималдық кодтау жүзеге асады.

Мысал. Бірөлшемді ДСТ мысалында ДСТ-ні түсіндірейік.

Екі өлшемді ДСТ – бұл бір өлшемді ДСТ жолдарынан алынған пиксельдердің блоктарының жолдарының әрбір қатары мен пиксельдердің блоктарының әрбір бағаналары үшін тізбекті түрде қолданылатын бірөлшемді ДСТ.

N іріктемелерге (кескіндегі пиксельдерге немесе дыбыстық файлдағы іріктемелерге) қолданылатын бір өлшемді ДСТ – бұл N ×N өлшемді матрица, оның жолдары мынандай косинустық функциялар болады:

$$DKT(m,n) = \sqrt{(1 - \delta(m,1)) / N} * \cos((\pi/N) * (n - 1/2) * (m-1)),$$

Мұндағы $m, n = 1, \dots, N$;

$p_i = 3.14159267\dots$;

N = блоктағы іріктеме саны (кескіндегі пиксельдер немесе дыбыстық файлдағы іріктемелер саны);

$DKT(m, n)$ – ДКТ-нің бұл бір өлшемді матрицасы;

Егер $m = 1$ онда $delta(m, 1) = 1$ әйтпесе $delta(m, 1) = 0$;

$sqrt(x)$ – түбір табу функциясы, $cos(x)$ радианмен өлшенеді.

Бұдан N іріктемеден тұратың блокқа ДСТ қолдану үшін көбейту мен қосудың $N*N$ операциясын орындау қажет болатындығын байқауға болады, алайда, ДСТ матрицасы рекурсиялық болғандықтан $N \log(N)$ операция ғана орындалады. Осы қасиетінің арқасында ДСТ заманауи дербес компьютерлердің математикалық процессорларында қолданылып отыр.

Дискреттік Wavelet-түрлендіру (DWT, Discrete Wavelet Transform). DWT қолданатын компрессорларға Intel Indeo 5.x; Intel Indeo 4.x жатады.

Артықшылықтары мен кемшіліктері:

– DWT алгоритмі арқылы сығылған статикалық және динамикалық кескіндердің көбінде ДКТ алгоритміндегідей блоктық құрылым болмайды.

– Бірдей сығу коэффициенттері болғанда DWT қолданып сығылған кескіндердің сапасы ДКТ алгоритмімен сығылған кескіндердің сапасынан жоғары болады.

– DWT-де әдетте кескінің контурларының бұлыңғыр, нақты болмауы, үшкір бұрыштарының ұштарының доғалдануы (*контурлық шу* немесе *Гиббс эффектісі* деп аталады) байқалады.

DWT-алгоритм сигналды (мысалы кескінді) төмен жиілікті және жоғары жиілікті фильтрлермен жіберуге негізделген. Төмен жиілікті фильтр бастапқы сигналдың жалпы нобайын (формасын) береді. Жоғары жиілікті фильтр DWT-алгоритм оны нақтылайды, айқын етеді (айырмашылық сигналын/ детализацияның қосымша сигналын береді). Өз кезегінде, жоғары жиілікті фильтрдің шығысындағы нәтижеге де (детализацияның қосымша сигналына) осындай осындай процедура орындалуы мүмкін және т.с.с.

DWT-алгоритм сигналдарды кодтау кезінде дискретті сигналдардағы артықтықты жоғалту мақсатында сығудың 1-ші кезеңінде қолданылады. Алғаш қарапайым DWT-ді венгр математигі Альфред Хаар 1909 жылы ойлап тапты. *Хаар*

түрлендіруі түрлі түсті және ақ-қара түсті суреттерді, әсіресе рентген түсірілімдерін жақсы сығады.

Ең кең тараған вейвлет-түрлендірулер топтамасын 1988 жылы бельгия математигі Ингрид Добеши ұсынды.

DWT-алгоритм мысалына Хаар вейвлетін қолданып 2^N өлшемді деректер массивін сығуға арналған бір өлшемді жылдам вейвлет түрлендірудің *Малл алгоритмі* жатады.

Ескерту: Толығырақ білу үшін мына ақпарат көздерін қараңыз:

https://ru.wikipedia.org/wiki/Вейвлет_Хаара,
<http://web.cecs.pdx.edu/~mperkows/CAPSTONES/HAAR/u9kvl.pdf>,
<http://arxiv.org/ftp/arxiv/papers/1010/1010.4084.pdf>.

Вейвлеттік сығудың қолданысы: Dirac, ECW, EZW, JPEG 2000, Motion Wavelet, MrSID, Pixlet, Kodudu, SIF1, Snow, SPIHT, StreamLogic, Tarkin, Windows HD Photo, ICER және т.б.

Кадрлар айырмашылығы технологиясы. Бұл технологияны қолданатын компрессорға Cinepak жатады.

Артықшылықтары мен кемшіліктері:

– Тәуелсіз жеке кадрларды сығуға қарағанда тұтас алғанда жақсы сығуды қамтамасыз ете алады.

– Кодтау барысында туындайтын қателер жинақталып, қосымша кілттік кадр болуын қажет етеді.

Кадрлар айырмашылығының алгоритмі көптеген бейнелерде кадрдан кадрға өту барысында кескіндердің бір-бірінен аса айырмашылығы болмау жағдайын қолданады. Векторлық квантизация алгоритмін әрбір келесі кадрды кодтау және қиын кодталатын аз коэффициенттерді алу үшін қолдану кезінде кадрларға біртіндеп қателер түседі. Бұл бейнеқатарға алдыңғы кадрларды ескермей кодталатын және бейнедегі «тіректік нүктелер» болып табылатын кілттік кадрлардың қосылуын талап етеді.

Қозғалыстың орнын толтыру технологиясы. Қозғалыстың орнын толтыру технологиясын қолданатын компрессорларға MPEG-1,2 және 4 жатады.

Артықшылықтары мен кемшіліктері:

– Кадрлар айырмашылығы технологиясымен салыстырғанда қозғалыстың орнын толтыру механизмі сығудың жоғары дәрежесіне қол жеткізеді.

– Кодтау күрделірек және арнайы аппаратураның болуын талап етеді.

– Қозғалыстың орнын толтыру технологиясы MPEG, H.261 және H.263 сияқты халықаралық цифрлық бейнені сығу стандарттарында қолданылады.

– Ең жоғары сығу баяу қозғалысты кадрларда жүзеге асады.

Қозғалыстың орнын толтыру технологиясы бірнеше күрделі алгоритмдерді қолдануға негізделген. Бұл технологияны бейнекадрлардағы объект қозғалмайтын фонға қатысты орнын ауыстыратын жағдайда қолданған тиімді. Ал формасы өзгертін немесе жақыңдалатын немесе алыстатылатын (қозғалмалы камера) объекттерді бұл алгоритм тиімді сыға алмайды. Негізгі блок ретінде (осыған қатысты қалған блоктардың ығысу векторы анықталады) кодталатын (болжамдалатын) блокқа қатты ұқсас, 16×16 пиксель өлшемдегі кез келген блок болуы мүмкін. Басқа кадрлар сілтеме жасайтын кадр алдын ала олардан бұрын декодталуы тиіс. Алайда, тіректік кадр болжамдалатын кадрдың алдында болуы міндетті емес. Мысалы, MPEG екі бағытта да В-кадрларды (bi-directionally predicted) енгізу арқылы болжамдай алады.

MPEG (Moving Picture Experts Group) - цифрлық бейнені бейнелеудің танымал форматы, MPEG-1, MPEG-2 және MPEG-4 сияқты 3 спецификациясы бар. MPEG сарапшылар тобының ресми атауы – ISO/IEC JTC1 SC29 WG11, бұл топ екі ұйымның – ISG (Халықаралық стандарттар бойынша ұйым) және IEC (Халықаралық электротехникалық комиссия) бірлескен жетекшілігімен жұмыс жасайды. Топтың міндеті – аудио және бейнесигналдарды кодтаудың бірыңғай нормаларын жасау. MPEG стандарттары CD және CD-Video технологияларында қолданылады, DVD бөлігі болып табылады, кабельді және жерсеріктік теледидарларда, ISDN арналары жүйелерінде, цифрлық радио арқылы хабар таратуда және көптеген әртүрлі электронды ақпараттық жүйелерде қолданылады.

(Ескерту: Толығырақ <http://www.mpeg.org/MPEG/> сайтына қараңыз)

MPEG-сығу келесі идеяларға негізделген:

– Бейненің (video) уақыттық артықтығын алып тастау – бұл қысқа уақыт аралығында сценаның көпшілік фрагменттерінің

козғалыссыз болуын немесе өрісте мардымсыз ғана ығысуын ескеруге негізделеді.

– *Кескіннің (image) кеңістіктік артықтығын алып тастау* – бұл адамның визуальды қабылдауы үшін маңызды емес болатын сценадағы ұсақ детальдарды қарастырмауға негізделеді.

– *Кескінді yuv-бейнелегенде (y – жарықтық (яркость), u мен v – түстері әртүрлі сигналдар) түстердің төмен шешімін (разрешения) қолдану* – бұл адам көзінің жарықтықтың өзгерісіне қарағанда түстердің рендерінің (оттенки) кеңістіктегі өзгерістеріне сезімталдығы аз болуына негізделеді.

– *Нәтижелік сығылған цифрлық ағынның ақпараттық тығыздығын арттыру* – бұл оны сипаттауға арналған тиімді математикалық кодты (мысалы, жиі қайталанатын мәндер үшін қысқа кодтық сөздерді қолдану сияқты) таңдауға негізделеді.

MPEG-сығудың басты ерекшелігі – кадрларды жеке редакторлеуге болмайды.

MPEG-1 – бұл негізі JPG-кодектен алынған әлемге танымал формат. Мұндағы сығу үш кадрдан тұратын сериялар бойынша жүзеге асады. Бұл – «әлсіз» машиналарда да бейнені стереодыбыспен осы форматта көруге мүмкіндік беретін ескі кодектерге жатады, соған орай кескіннің сапасы да жоғары емес, оның сапасы аналогтік VHS форматымен теңдес деуге болады. Суреттің өлшемі 352x288 нүкте, сапасы аса жақсы емес, қазіргі компьютерлер мен интернеттегі деректерді жіберу сыйымдылығы мен жылдамдығының қарқынды дамуына байланысты MPEG-1 форматы қолданыстан шыға бастады.

Бұл *MPEG-1 форматында деректерді сығу* келесі түрде жүреді. Айталық, бейнекадрда көлік «А» пунктiнен «Б» пунктiне қозғалып келе жататын сцена бар болсын. Машинаның орын ауыстыруын екі параметрмен сипаттауға болады: «А» пунктiнен «Б» пунктiне векторлық орын ауыстыру және өз осiнен айналу бұрышы. Артқы фон бұл жағдайда өзгерiссiз немесе өзгермейтiн дерлiк болады, әдетте көрермен фондағы алыстан көрiнетiн ағаштардың ұсақ бұтақтарының қозғалысына назар аудармайды. Сондықтан, кадрды екі құраушы бөлiкке бөлiп тастауға болады: 1) бiр рет қана сақталатын және содан соң барлық кадрдың жаңғыртылуында қайта көрсетiлетiн артқы фон және 2) көлiктiң қозғалатын аумағы – оны әр кадр үшін жеке жазу керек болады.

MPEG-1 форматында бейнероликтің барлық кадрлары, яғни MPEG-тізбектер I-кадр, P-кадр және B-кадрлары болып үш типке бөлінеді.

I-кадрлар (Intra Frames) – тіректік кадрлар, осылар арқылы айырмашылықтары бойынша басқа кескіндер бастапқы калпына келтіріледі. Олардың кескіндері толық көлемде JPEG форматында сақталады.

P-кадрларға (Predicted Frames) ағымдағы кескіннің алдыңғы болған I-кадрдан айырмашылығы жазылады, нәтижесінде ең аз дисктік кеңістікті қажет етеді.

B-кадрларға (Bi-directionally interpolated Frames) ағымдағы кескін үшін алдыңғы және келесі болатын I-кадр немесе жеке фрагменттердің ығысуын ескергендегі P-кадр типіндегі кескіндердің арасындағы айырмашылықтар жазылады.

Нәтижесінде сығылған файл өлшемі шамамен бастапқының 1/35 бөлігін құрайды. Алайда, Internet немесе жерсеріктік хабар тарату желілері арқылы бейнені сығып жіберуге бұл стандарт жарамайды.

MPEG-2 жоғарыда аталған MPEG-1 форматының кеңейтілуі болып табылады. Мұнда ұсынылатын кадр өлшемі жоғарылаған, ол 1920×1080 нүктелерді құрайды, алты арналы дыбыстық қолдау қосылған. Алайда сығылған бейнені аталған форматта жаңғырту үшін компьютердің жоғары есептеу қуаттылығы қажет етіледі.

Қазіргі кезде басым қолданыстағы формат – бұл 720×576 нүктелік кеңейтілуі бар MPEG-2 форматы. Барлық DVD-video дисктері MPEG-2 форматында жұмыс жасайды. Сығудың MPEG-2 форматын бірнеше арналарға бір жиілікте жерсеріктерден трансляция жасағанда, сонымен қатар эфирлік трансляцияда, соның ішінде жоғары айқындылықты теледидарда, әртүрлі DVD-плеерлерде, microMV-бейнекамераларда қолданылады. MPEG-2 арқылы стандартты DVD (4,7 Гб) дискке бір жарым сағаттық сапасы жоғары фильмді жазуға болады. Сонымен қатар, бұл форматта бірнеше әртүрлі дыбыс жолдары (дубляж), көпарналы дыбысталудың әртүрлі форматтарын, субтитрларды, бейнематериалды шолудың әртүрлі бұрыштарын (бірнеше синхронды бейне жолдары) және басқа да жаңа мүмкіндіктерді қолдана отырып қосарлы DVD (9 Гб) дисктерге де жазу мүмкіндігі

бар. Соның нәтижесінде, дисктегі бейнематериалдың кез келген бөлігіне жылдам еркінше қатынау мүмкіндігі бар.

MPEG-2 форматы 1920x1080 пиксельдерге дейінгі кеңейтуді қолдануға мүмкіндік береді (өріспен немесе өріссіз, секундына 25 кадр) және 6 арналық дыбысты қолдайды.

MPEG-2 форматының ерекшеліктері – деректер ағыны онда 12 Мбит/с шамасына артуына (DVD-дегі 9,8 Мбит/с максимал стандартымен салыстырғанда), ал кассета мөлшері төмендетілгеніне (DV стандартымен салыстырғанда) қарамастан Sony компаниясы өзінің кеңейтілген microMV стандартында кең қолданады. Сонда да DV стандарты жоғары тұрақтылығымен, әлем бойынша кең таратылуымен ерекшеленеді.

MPEG-2 форматындағы бірден miniDVD дисктерде жазатын камералар бар. Олардың маңызды артықшылықтары: сапасын жоғалтпай дисктерді 1000 ретке дейін қайта жазу, материалдарының қол жетімділігі және басқа да өзгешеліктері бар. Әуесқойлық түсірілімдер үшін бұл тиімді, миниDVD дисктер тұрмыстық плеерлер мен компьютерлерде жақсы жаңғыртылады, ал мұндай камералармен жүретін программалар DVD-келтіргішпен жабдықталған кез-келген компьютерде бейнеге монтаж жасауға мүмкіндік береді. Кемшілігіне нақты жазылатын материал көлемінің (1 mini DVD дискте 30 минутқа дейін) шектеулі болуы жатады.

MPEG-3. MPEG-3 стандартын құрастыруға қатысты жұмыстар жасалып, деректер ағынының жылдамдығы 20–40 Мбит/с жоғары айқындылықты теледидар жүйесінде (high-definition television, HDTV) қолдануға негізделген болатын, алайда кейінірек ол MPEG-2 стандартының бөлігі болып қалды, қазір бөлек айтылмайды.

Ескерту: MPEG-3 форматыш дыбыстық деректерді сығуға арналған MP3 форматымен шатастырмау керек! MP3-тің толық атауы –MPEG Audio Layer III.

MPEG-4. MPEG-4 – бұл цифрлық дыбыс пен бейнені сығуға арналған халықаралық стандарт, 1998 жылы пайда болды. Құрамына дыбыс мен бейнені сығу стандарттары және аралас технологиялар бар (ISO стандарттау бойынша халықаралық ұйым/IEC Moving Picture Experts Group (MPEG) мақұлдаған), оған MPEG-1, MPEG-2 және т.б. стандарттардың көптеген функциялары кірген.

MPEG-4 форматы негізінен дыбыс пен бейнені сығу үшін интерактивтік мультимедиада, цифрлық теледидарда, графикалық қолданбаларда қолданылады, сонымен бірге компакт-дисктерге фильмді жазу үшін де, бейнетелефонияда да, «мультимедиалық контент» құру үшін де (интерактивтік теледидарды, 3D-кескіндерді, мәтіндерді және т.б. біріктіру үшін) бағдарланған.

MPEG-4 форматымен сығылған бейненің сапасына бейнероликтің бастапқы сапасы, сығу параметрлері және компьютердің техникалық сипаттамалары әсер етеді.

MPEG-4 форматының артықшылықтары: сығу нәтижесіндегі бейне файлдардың өлшемі шағын болады (мысалы, 4 Гб DVD форматынындағы фильмді осы форматпен кодтағанда 700 Мб болады), қарауға тиімді, жазу сапасы жақсы, сондықтан интернетте бейнені жіберу үшін аса жиі қолданады.

MPEG-4 форматының кемшіліктері: DVD форматына қарағанда сығылған дыбыс пен бейненің сапасы төмендеу болады (алайда, DVD форматындағы бейненің өлшемі MPEG-4 форматындағыға қарағанда үлкен болатынын ескеріңіз!).

Жақсы экранда анықтап қараса MPEG-2 пен MPEG4 форматтарында кодталған кескіндер арасындағы ерекшелікті байқауға болады. Алдыңғы форматтарға қарағанда дыбыстың сапасы мен бейнесигналдың максималды тығыздығы 30-40 % артық. Айырмашылығы тізбектегі кадрлар саны 3-тен артық (250 дейін), нәтижесінде сығылуы жоғары болғандықтан интернетте нақты режімде сапалы бейнені көруге, компакт-дискке 1 жарым сағаттық сапасы жақсы бейнені жазуға болады.

Форматтың маңызды ерекшелігіне типтік нысандар үшін олардың қозғалыстарын сипаттауға және болжамдауға арналған (атап айтқанда адамдар жүрісі, ең көп таралған ымдасулар, мимикалар) жеке алгоритмдердің жасалғанын айтуға болады. Кадрдағы мұндай өзгерістерді жазу қажет емес болып қалды, өйткені оларды программада есептеуге болады.

MPEG-4 форматында бейнекескінің үстінен әртүрлі шрифтітермен жазылған мәтінді көрсетуге, тіпті бұл мәтінді сөйлеу синтезаторымен әйел және ер адам дауыстарын имитациялау арқылы дыбыстауға болады. Қажет жағдайда дыбысталатын фонемаларға сәйкес дыбысталу диктор бетінің қозғалысымен синхрондалады. Сонымен қатар кейбір музыкалық аспаптардың

дыбысталуын синтездеуге болады. Цифрленген дыбыстық жазбалар арнайы жасалған AAC кодек (Advanced Audio Codec) арқылы тиімді сығылады.

Кейбір бейнекамералар бейнені MPEG-4 форматында өзінің жадтық қартасына жаза алады немесе USB кабель арқылы дыбысы бар бейнені MPEG-4 форматында жіберу арқылы web-камера ретінде жұмыс жасай алады. Қазіргі заманғы технологиялар GPRS колдана отырып мобилді телефондар арқылы цифрлық (MPEG-4 немесе MPEG-2 форматында қысқартылған) теледидарды жаңғыртуға мүмкіндік береді.

Қазір де MPEG-4 – бұл бейнені интернетте және дербес компьютерлерде таратудағы ең танымал формат. Бейненің сапасы жақсы болған жағдайда жадты ұтымды пайдалану тиімді болып табылады. Әрбір келесі MPEG-4 кодек версиясы (қазіргі кезде 3.xx, 4.xx және 5.xx версиялары) жаңа озат өрістерге дәніктіреді. Тұрмыстық плеерлер және басқа да көптеген құрылғылар мәселесіз бұл форматпен жұмыс атқарады. MPEG-4 алдағы уақытта оның орнына жаңа формат келмейінше әлі де маңызды болады.

AVI-файлдары – мультимедиалық деректермен алмасуға арналған бұл формат RIFF (Resource Interchange File Format сөзінен қысқартылған) файлдарының ерекше жағдайы. Аталған формат жеке компьютерлердегі сығылған бейнені қалпына келтірудің кең таралған формасы болып табылады. Бейнемәліметтерді келтіру қалпына байланысты AVI-файлдары әртүрлі стандартты болып табылады.

Editable MPEG компрессоры і типті MPEG кадрларынан құралған AVI-файл, *M-JPEG* сияқты сандық бейнені реттеуде кең қолданылады. Алайда MPEG форматының барлық басқа да сығу механизмдері осында жасақталған. Microsoft Video for Windows 1.1. стандарттының құрамына кіреді және Adobe Premiere сияқты сандық бейнені реттеу жүйесімен қолданылады.

Жоғарыда келтірілген компрессорлар танымал болғанымен, бұл AVI-файлдарының барлық стандарттары емес. Компрессорлардың бұл тобын сипаттай отыра ең алдымен қатты диск және компакт-дисктерде сақталатын, ал бұл өз кезегінде жаңғыртудың жоғары сапасына қатысты олардың сығу кезіндегі мүмкіндіктерінің шектеулілігін білдіретін бейне және аудиодеректерді сығу құралы ретінде жасалғандығын ескерген жөн.

Интернеттің арқасында алдыңғы қатарлы технологияларды (sophisticated motion estimation and compensation, wavelets, fractals және басқалары) қолдана отырып бейне және аудиодеректерді сығу құралдары (мысалы, Интернет құралдары арқылы бейнеконференция сеанстары) аса танымал болып отыр. Сығудың мұндай түрлері төмен сапада сығудың жоғары дәрежесін қамтамасыз ете алады.

MJPEG [Motion JPEG] – бейнені кодтау стандарттары тобының ресми емес атауы. MJPEG компрессия әр кадрды тәуелсіз кодтап, алынған тізбекті файлға біріктіруге негізделген.

Бейнені сығу JPEG-алгоритмі бойынша жүреді: әрбір сурет (кескін) 8x8 нүктелік торларға бөлінеді, дискреттік түрлендіру мен алыныған спектрді жоғары жиілікті фильтрациялау арқылы векторлық түрге келтіріледі. Басқаша айтқанда, сығылған бейне JPEG-кескіндердің тәуелсіз тізбегі түрінде көрсетіледі, кадр арасындағы айырмашылықтар ескерілмейді.

Бейнені кодтаудың танымал форматы MJPEG жоғары сапалы стоп-кадрлар беретіндіктен, цифрлық фотоаппараттарда, веб-камераларда, сызықты емес бейнемонтаждарда, IP-камераларда, бейнебақылау жүйелерінде қолданылады, сондай-ақ QuickTimePlayer, PlayStation консоліне, Google Chrome, Safari, Mozilla Firefox браузерлеріне кіріктіріліп қолдау тапқан.

MJPEG алгоритмінің артықшылығы – оның симметриялы болуы, яғни кодтау мен декодтау үшін бірдей есептеу шығындары жұмсалады, MPEG-4 алгоритміне қарағанда есептеу ресурстарын аз қажет етеді және аппараттық жүзеге асуы да арзанырақ. Бұл форматты бейнені редакторлеу үшін қолдану тиімді, өйткені әр кадр басқаларға тәуелсіз кодталғандықтан, әр кескінді жеке редакторлеуге болады.

MJPEG арқылы бейнені сығу дәрежесі 1:15, нәтижесінде бейнеаппаратты сапасын жоғалтпай сақтауға болады, 1:15 пен 1:25 аралығында болса айқындық шегі (разрешение) аздап жоғалады.

MJPEG кемшіліктері:

– Сығу дәрежесі 1:30-дан жоғары болғанда MJPEG алгоритмімен бейнені сығу кезінде JPEG форматындағы кездесетін бұрмаланулар болады, атап айтқанда: 8x8 нүктелік торлардың шекарасында кескіннің тегістігі бүлініп, шеттері «мозаика» тәріздес болады.

– Сығу тиімділігі аса жоғары емес, уақыттық артықтықты жою қарастырылмағандықтан MPEG алгоритмдеріне қарағанда сығу дәрежесі төмен (*Ескерту*: Екеуінің салыстырулары туралы мына дереккөзден http://videocodec.info/books/MJPEG_vs_Mpeg_4.pdf қараңыз).

– Өлшемдері 2 Гб артық бейнефрагменттер жасалынбайды, файл құрылымы оның өлшемін арттыруға мүмкіндік бермейді. Қазіргі кезде MJPEG-файлдарды біріктірудің («желімдеу») программалық әдістері қолданылатындықтан, фрагменттер арасындағы өтулер байқалмайды.

– MJPEG-тің барлығына арналған формат спецификациясының бірыңғай құжаты болмағандықтан, әртүрлі өндірушілер мен әзірлеушілер жасаған MJPEG-тің жүзеге асырулары кездеседі, тіпті кейбіреуі бір-біріне сәйкес келмей жатады.

MJPEG компрессиясы бірнеше жылдар бұрын мультимедиа саласының стандарты болғандықтан, кезінде жеке MJPEG-кодектер де жасалынған. MJPEG форматы JPG стандарты (768x576 нүктелік) бойынша аналогтік бейнесигналдың қарапайым өңделуін қолданады, ал декомпрессиясы Motion-JPEG (қозғалатын JPEG) түрінде болады. Сығылған сапалы бейнероликтер көп орын алатындықтан, қазіргі кезде бұл формат аса қолданылмайды, тек кейбір бейне функциялары бар фотокамераларда, бейнебақылау жүйелерінде шешімі (разрешение) 320x240 нүктелік M-JPEG қолданылады.

JPEG (Joint Photographic Experts Group – фотосуреттер бойынша біріккен сарапшылар тобы) – әзірлеуші ұйым атауымен аталған, бұл комитет ISO құрамына кіреді.

JPEG – бұл сонымен қатар фотолар мен суреттерді сақтауға арналған (файл кеңейтілімдері .jpg, .jfif, .jpe, .jpeg), танымал графикалық формат, ең танымалысы – jpg форматы.

JPEG сығу алгоритмі бастапқыда жадта аз орын алу үшін статикалық (қозғалмайтын) суреттерді сығу үшін қолданылды. JPEG сығу алгоритмі веб-камераларда, бейнесерверлерде, кейбір желілік құрылғыларда (мысалы, AXIS Communications компаниясының веб-камералары мен желілік бейнесерверлері, JVC Professional компаниясының VN-A1U бейнесервері), көбінесе цифрлық фотосуреттерде, суретті сақтау және Интернет желісі арқылы жіберу үшін қолданылады.

JPEG сығу алгоритмінің жұмысы дискретті косинустық түрлендіруге (DCT, Discrete-Cosine Transform) негізделген.

Кодтау процесі келесі әрекеттерді орындаудан тұрады:

- суретті 8x8 пикселдік блоктарға бөлу;
- әр блокта жылдам дискретті косинустық түрлендіруді орындау (жылдам DCT);
- алынған DCT коэффициенттерін кванттау коэффициенттерінің кестесін қолдана отырып кванттау (Қкестесі);
- суреттің әрбір блогының DCT-нің квантталған коэффициенттерін энтропиялық кодтау.

Кодталған деректерді бейнелеудің 2 тәсілі бар: *тізбекті* және *прогрессивті JPEG* (sequential JPEG, progressive JPEG), екеуі де DCT-ге негізделгендіктен декомпрессиялағанда бастапқымен бірдей болатындай етіп суретті қалпына келтіре алмайды.

DCT қолданған жағдайда әрқашан шығынмен (жоғалтуы бар) сығу болатындықтан, JPEG деректерді аз жоғалта отырып, аса жоғары сығу дәрежесін береді. Түрлі түсті суреттерді 5-10 есе көріну сапасын жоғалтпай сыға алады.

Алайда, көрші пикселдер арасындағы күрт өзгеретін контраст нәтижесі көзге байқалатын бұрмалануларға әкелетін, яғни жарықтықтың (яркость) екі деңгейі ғана болатын суреттерді (сызбалар, мәтіндік және символдық графика) сығуға, сондай-ақ астрономиялық және медициналық суреттерді сығуға бұл алгоритмді қолдануға болмайды, оларды шығынсыз сығатын JPEG-LS, TIFF, GIF, PNG сияқты форматтарда сақтау керек.

Формат ұғымы. Ескерту. Бөлімде қолданылған «формат» ұғымы мен «сығу алгоритмі» ұғымы бір емес, синоним де емес. Мысалы, RLE сығу алгоритмінің әртүрлі модификациялары аса көп форматтарда, соның ішінде TIFF, BMP, PCX форматтарында жүзеге асқан.

Егер де қандай да бір форматтағы файлдың өлшемі үлкен болса, бұл сығу алгоритмінің «нашарлығын» білдірмейді, берілген кескін үшін бұл форматта қолданылған алгоритм нашар нәтиже көрсеткенін білдіреді.

Сонымен қатар, көптеген заманауи форматтар сығу алгоритмдерін қолданып та, қолданбай да дискке жазуды қолдайды. Мысалы, TIFF 6.0 форматы кескінді RLE-PackBits, RLE-CCITT, LZW, бекітілген кестесі бар Хаффман алгоритмі, JPEG

алгоритмдерін қолданып та, ешқандай сығу алгоритмдерін қолданбай да сақтай алады. Сол сияқты, BMP мен TGA форматтары файлды RLE алгоритмін (барлық модификацияларын) қолданып та, қолданбай да сақтайды.

PC, Apple, UNIX платформаларында жиі қолданылатын форматтардың тізімі: ADEX, Alpha Microsystems BMP, Autologic, AVHRR, Binary Information File (BIF), Calcomp CCRF, CALS, Core IDC, Cubicomp PictureMaker, Dr. Halo CUT, Encapsulated PostScript, ER Mapper Raster, Erdas LAN/GIS, First Publisher ART, GEM VDI Image File, GIF, GOES, Hitachi Raster Format, PCL, RTL, HP-48sx Graphic Object (GROB), HSI JPEG, HSI Raw, IFF/ILBM, Img Software Set, Jovian VI, JPEG/JFIF, Lumena CEL, Macintosh PICT/PICT2, MacPaint, MTV Ray Tracer Format, OS/2 Bitmap, PCPAINT/Pictor Page Format, PCX, PDS, Portable BitMap (PBM), QDV, QRT Raw, RIX, Scodl, Silicon Graphics Image, SPOT Image, Stork, Sun Icon, Sun Raster, Targa, TIFF, Utah Raster Toolkit Format, VITec, Vivid Format, Windows Bitmap, WordPerfect Graphic File, XBM, XPM, XWD.

Ескерту. Бейне ақпаратты және дыбыстық ақпаратты сығу, қалпына келтіру (компрессия, декомпрессия) әдістері, стандарттары туралы толығырақ білу үшін мына әдебиеттерді қараңыз:

1) Сэломон Д. Сжатие данных, изображений и звука. – М.: ТЕХНОСФЕРА, 2004 [21];

2) Ватолин Д., Ратушняк А., Смирнов М., Юкин В. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео. – М.: ДИАЛОГ-МИФИ, 2002. – 384 с. [4];

3) В.В.Лидовский, Основы теории информации и криптографии [3];

4) Ливак Е.Н. Алгоритмы сжатия. <http://mf.grsu.by/UchProc/livak/po/comprsite>.

5) Бейнені кодтаудың танымал стандарттары H.120, H.261, H.263, H.264, MPEG-1, MPEG-2, MPEG-4 туралы <http://videocodec.info/>, http://videocodec.info/books/1998_image_and_video_coding_emerging_standards_and_beyond.pdf

3.6 Дыбыстық ақпаратты сығу әдістері

Цифрлық аудиоформат – компьютерде немесе дыбыстық тасымалдауыштарда сақтау үшін цифрлық дыбыстық жазуда қолданылатын дыбыстық (аудио) деректерді бейнелеу форматы.

Аудиокодек – дыбыстық деректердегі артықтықты жою үшін аудиодеректерді сығуды іске асыратын кодер мен декодер.

Деректерді жоғалтып немесе жоғалтпай кодтауға байланысты аудиоформаттарды әдебиеттерде 3 негізгі топқа бөледі:

– сығылмаған аудиоформаттар (AIFF, WAV, CDDA, DSD, DXD...);

– шығынсыз сығуы бар аудиоформаттар (ALE, FLAC, TAK, TTA, Shorten, Windows Media Audio 9 Lossless...);

– шығынмен сығуы бар аудиоформаттар (MP2, MP3, Ogg, Windows Media Audio, AAC,...).

Бұлардан басқа да көптеген форматтар, плейлистердің нақты медиаплеерлерге арналған дыбыстық форматтары бар.

Кодтауы мен декодтауы өте жылдам жүретін аудиоформаттарға Shorten (сығу дәрежесі 60,9%), TAK (сығу дәрежесі 53,9%), ал баяу кодтайтын, декодтайтындарға LA (сығу дәрежесі 52,1%, ең тиімді сығу), OptimFRGG (сығу дәрежесі 53,2%) жатады. Ал, WavPack тек өте жылдам кодтаса, FLAC тек өте жылдам декодтайды, олардың сығу дәрежелері бірдей – 55,7%.

Кейбір танымал аудиоформаттардың сипаттамаларын қарастырайық.

Apple Lossless (Apple Lossless Encoder, ALE немесе **Apple Lossless Audio Codec, ALAC** деп танымал) – бұл Apple Inc жасаған кроссплатформалық аудиокодек цифрлық әуенді сапасынан айырмай сығуға мүмкіндік береді. Apple Lossless деректері MP4 контейнерінде **.m4a** кеңейтілімімен сақталады. Алгоритмнің кодтауы мен декодтауы жылдам.

Apple Lossless форматының кеңейтілімі AAC форматы сияқты болғанмен, бұл AAC емес, бұл кодек FLAC сияқты және т.с.с. lossless-кодектерге ұқсас. iPod плеерлері (shuffle емес) Apple Lossless форматындағы дыбыс форматтарын жаңғырта алады. Кодекте цифрлық басқару құқықтарының (DRM) қандай да бір өзіндік құралдары қолданылмайды, бірақ DRM-ді қолдану

контейнер форматымен қарастырылғандықтан мүмкін болып саналады.

Сығылған ALAC файлдарының өлшемі басқа да lossless-форматтар сияқты әуен түріне тәуелді болатын түпнұсқа өлшемдеріне байланысты шамамен 40%-дан 60%-ға дейінгі мөлшерде болады. Мүмкін болатын декодталу жылдамдығы оны iPod сияқты өнімділігі шектеулі құрылғылар үшін қолдануға тиімді етеді.

Apple Lossless Encoder iTunes 4.5. функциясы ретінде де QuickTime 6.5.1 құраушыларының бірі ретінде (сәуір, 2004 жыл) ұсынылды. Аудиокодек сонымен бірге iTunes жүзеге асырылуында AirPort Express-те де қолданылады.

Apple Lossless форматына арналған декодер аудиобейне файлдарды кодтау мен декодтауға арналған libavcodec атты әмбебап кітапханада бар. Кез келген мультимедиалық ойнатқыш (соның ішінде VLC media player мен MPlayer де) осы аталған кітапхана негізінде Apple Lossless-файлдарды жаңғырта алады.

MPEG-4 ALS (MPEG-4 Audio Lossless Coding) – NTT корпорациясының бұл кроссплатформалық өнімі дыбыстық сығуды шығынсыз жүзеге асыратын MPEG-4 стандартының кеңейтілуі болып табылады. MPEG-4 ALS – танымал формат. 2006 жылы ISO/IEC 14496–3:2005/Amd 2:2006 ретінде ұсынылды. Алгоритмнің кодтауы орташа жылдамдықпен, ал декодтауы жылдам жүреді, 55,1% -ға дейін сығады.

FLAC (Free Lossless Audio Codec) – дыбысты шығынсыз сығуға арналған танымал еркін кроссплатформалық аудиокодек, аудиофайлдардың форматы (Xiph.Org Foundatio өнімі, сайты flac.sourceforge.net), файл кеңейтілімі – .flac. Қазіргі кезде аудиоплеерлерде, арнайы аудиожүйелерде кеңінен қолданылады, әртүрлі аппараттық жүзеге асулары да бар.

Сығуда шығындары бар Ogg Vorbis, MP3, AAC сияқты аудиокодектермен салыстырғанда FLAC басқа да lossless-кодектер сияқты аудиоағыннан ешқандай ақпаратты өшірмейді және дыбысты жаңғыртатын аппаратурада жоғары сапалы әуенді тыңдау үшін де, аудиотоптаманы архивтеу үшін де қолданылады.

MP3 (MPEG-1/2.5 Layer 3) – MPEG дыбыстық жолын кодтаудың форматының үшінші деңгейі, дыбыстық ақпаратты

сақтауға арналған файлдың лицензияланған форматы. (*Ескерту: MPEG-3 емес!*)

Қазіргі кезде MP3 форматы – дыбыстық деректерді шығынмен цифрлық кодтаудағы кеңінен таралған форматтардың ішінде ең танымалы, барлық операциялық жүйелерде көптеген портативті аудиоплеерлермен, DVD-плеерлермен ашылады. Ол файлдармен алмасу желілерінде аудиофайлдардың жіберілуін бағалау үшін кеңінен қолданылады.

MP3 форматы екі арналық кодтауды (стерео) қолдайтындықтан, төрт режимі бар:

- *стерео* – екі арналық кодтау,
- *моно* – бір арналық кодтау,
- *екі арналық стерео* – екі тәуелсіз арна үшін (әртүрлі тілдердегі дыбыстық сүйемелдеулер),
- *біріктірілген стерео* – екі арналық кодтаудың тиімді тәсілі.

Ең көп таралған MP3-тің кодектеріне mp3PRO-codec, LAME-codec, fraunhofer-codec жатады.

MP3 форматында жазбаны жаңғырту және түпнұсқасына жақын болатындай етіп жаңғырту сапасын қамтамасыз ету мақсатында деректердің өлшемдерін нақты кішірейтуге арналған шығындармен сығу алгоритмі қолданылады. Орташа битрейті 128 кбит/с болатын MP3 жасаған кезде нәтижесінде аудио CD-ден түпнұсқа файлдан өлшемінің шамамен 1/10 мәніне тең болатын файл шығады. MP3 файлдары нәтижелік сығылған файлдың сапасына әсер ететін жоғары немесе төмен битрейтпен құрылуы мүмкін. Сығу алгоритмі көптеген адамдардың есту мүшелеріне аса анық білінбейтін, кейбір дыбыстық ағындардың бөліктерінің нақтылығын төмендетуге негізделген. Бұл тәсіл *қабылдауды кодтау* деп аталады, оның түпнұсқадан айырмашылығы жоғары нақтылығы бар аппаратурада ғана білінеді. Бастапқы кезеңде уақыттың қысқа аралықтары ретіндегі дыбыс диаграммасы құрылады, содан соң адамның есту мүшесіне анық естілмейтін ондағы ақпарат өшіріледі, ал қалған ақпарат жинақы түрде сақталып қалады. Бұл тәсіл JPEG форматындағы кескіндерді сығуда қолданылатын тәсілге ұқсас.

Бұл форматта дыбыстар жиіліктік сипатта кодталады (дискретсіз партиясыз); екі форматта болатын стерео кодтауы бар. MP3 шығынмен сығу форматы болып табылады, яғни адам құлағы қабылдай ала алмайтын (психоакустикалық үлгіге сәйкес) немесе

тек кейбір адамдармен ғана қабылданатын дыбыстық деректердің бөлігі жазбадан қайтарымсыз өшіріледі. Сығу дәрежесін бір файл шегінде түрлендіруге болады. Битрейттің мүмкін мәндерінің аралығы 8–320 кбит/с кұрайды. Дискретизация жиілігі 44100 Гц және 16 бит кұрайтын қарапайым компакт-дисктің Audio-CD форматынан деректер ағынын салыстыру үшін кванттау шамамен 1411 кбит/с.

Әртүрлі қажеттіліктер үшін MP3 форматтың битрейт пен дискретизация жиілігінің мүмкін диапазондарымен ерекшеленетін үш нұсқалары (MPEG–1, MPEG–2 және MPEG–2.5) бар:

- 1) 32000 Гц, 44100 Гц және 48000 Гц дискретизация жиіліктері MPEG–1 Layer 3 үшін 32–320 кбит/с болады;
- 2) 16000 Гц, 22050 Гц және 24000 Гц дискретизация жиіліктері MPEG–2 Layer 3 үшін 16–160 кбит/с болады;
- 3) 8000 Гц және 11025 Гц дискретизация жиіліктері MPEG–2.5 Layer 3 үшін 8–160 кбит/с болады.

MP3-файл бірнеше MP3 фрагменттерінен (фреймдерінен), ал әр фрагмент деректер блогы мен тақырыбынан тұрады. Бұндай фрагменттердің тізбегі *қарапайым ағын* деп аталады. Фрагменттер тәуелсіз элементтер («байт резервуары») болмағандықтан оларды еркінше шығарып алу мүмкін емес. MP3 файлдың деректер блогы жиіліктер мен амплитудалар түріндегі сығылған аудио деректерден тұрады. MP3 тақырыбында дұрыс MP3 фрагментін табуға арналған маркер болады. Содан кейін тұрған бит MPEG-тің қолданылуын, ал келесі екі бит MPEG–1 Audio Layer 3 немесе MP3 қолданылып тұрғандығын көрсетеді. Келесі мәндер MP3-файл түріне байланысты әртүрлі болады. Тақырыптың әрбір секциясының мәндерінің диапазоны мен спецификациялары ISO/IEC 11172–3 стандартында анықталған.

MP3 форматының кемшіліктері:

- 1) техникалық кемшіліктеріне дыбыс арналарының саны тек екеумен шектелгені жатады;
- 2) заңдық шектеулері бар: MP3 патентіне Alcatel-Lucent компаниясы ие болғандықтан оның меншігін заңсыз қолдануға болмайды, тек патенттің қызмет ету мерзімі біткен кезде MP3 форматын ақысыз қолдануға болады.

AAC (Advanced Audio Coding) – дыбыстық ақпаратты кодтаудағы кең жолақты алгоритм, жоғары сапалы цифрлық

аудионы жіберуге арналған деректер санын аса көп төмендету үшін екі негізгі кодтау принципін қолданады. Кодтау кезінде сигналдың адам қабылдамайтын бөлігі және кодталған аудиосигналдағы артықтық жойылады, содан кейін күрделілігіне сәйкес модификацияланған дискретті косинустық түрлендіру әдісімен сигнал өңделеді, содан кейін сигналға ішкі қателерді түзету коды тіркеледі де сигнал сақталады немесе арнаға жіберіледі.

AAC – аса сапалы, шығынмен сығуды қолданатын, қазіргі құрылғылардың көпшілігінде, соның ішінде портативті жабдықтарда қолданылатын формат. Алайт iTunes Store-ден әуен тек осы форматпен ғана жүктеледі.

AAC – бірдей битрейтері бар кодтау барысында MP3 форматына карағанда дыбыстық деректердің сапасын аз жоғалтып, аз шығынмен сығуды жүзеге асыратын патенттелген аудиофайл форматы.

AAC бастапқыда кодтау сапасы жақсартылып MP3 ізбасары ретінде жасалса, AAC ресми түрде ISO/IEC 13818–7 ретінде белгілі, 1997 жылы MPEG–2 жарыққа шықты және MPEG–4 Part 14 стандартында анықталған MPEG–4 ретінде белгілі AAC форматы бар. AAC файлының кеңейтілімі – .aac, MP4 контейнеріндегі кеңейтілімі – .mp4. Apple компаниясы аталған стандартта қарастырылмаған AAC файлдарының .m4a, .m4b, .m4p, .m4r кеңейтілімдерін қолданады.

AAC артықшылықтары: дыбыстық арна 48-ге дейін, тұрақты және өзгермелі битрейттерде жоғары тиімділікпен кодталады, дискретизация жиілігі 8–96 кГц (MP3-те 8–48 кГц ғана), жоғары жиіліктер өшірілмейді (MP3-те тек қатты дыбыстар сақталады).

WMA (Windows Media Audio) – көптеген аппараттық плеерлерде қолданылатын, аудиодеректерді сақтау және сығу үшін Microsoft компаниясы жасаған файлдың лицензияланған форматы, файл кеңейтілімі – .wma. Бастапқыда WMA форматы MP3 форматына балама ретінде болса, қазір MP3 форматының алдына шығып, сығу дәрежесі жағынан Ogg Vorbis және AAC форматтарымен бәсекелесе алады.

Қолданушылардың көпшілігі WMA форматының кемшілігіне қатеге төзімділігінің төмендігін жатқызады, яғни кодтау немесе жіберу кезінде бүлінген файлды ашу (жаңғырту) мүмкін болмайды. Ал, MP3 файлы зақымдалған жағдайда оны толығымен басынан

бастап зақымдалған жерге дейін жаңғыртуға, содан соң бірнеше секунд жберіп алып, оны ары қарай жаңғыртуға мүмкіндік береді; кейде MP3 файлдарындағы бірнеше байттағы қателер тындау кезінде аз байқалады немесе тіптен білінбеуі де мүмкін. сапасы тиімді бола береді жеп болжамдауға болады.

Көптеген портативті аудиоойнатқыштар MP3 форматымен тең қатарлы WMA форматын қолдайды. WMA жабықтылығы салдарынан басқа платформаларда қолдау ташпаған.

WMA форматына Microsoft авторлық құқықтарды басқарудың цифрлық жүйесін (DRM, қорғау жүйесі) қосты. Соның нәтижесінде қорғалған файлды басқа компьютерлерде тындауға мүмкіндік болмайды, әрі композиция әуендік дүкеннен жазылып алынуы тиіс. Windows Media Audio 9.1 версиясынан бастап форматтың соңғы түрлерінде сапасын жоғалтпай кодтау (*lossless*), көлемді дыбысты кодтау және дауысты (*voice*) кодтау қарастырылған.

БӨЛІМ БОЙЫНША БАҚЫЛАУ СҰРАҚТАРЫ:

1. Графикалық ақпараттарды (image, video) сығуда олардың қандай ерекшеліктері ескеріледі?
2. Кескіндер (image, сурет) шартты түрде қалай жіктеледі?
3. Графикалық ақпараттарды сығу сапасының қандай деңгейлері (сығу дәрежелері) бар? Мысалдарын келтіріңіз.
4. Графикалық ақпаратты сығу алгоритмдері жүйелердің қандай шартты түрде 3 тобында қолданылады?
5. Сығу алгоритмдерін салыстырғанда қандай жағдайлар мен талаптар ескеріледі?
6. Кескінді сығу алгоритмдерінің қандай түрлері бар?
7. Цифрлық бейнені сығудың қандай технологиялары бар?
8. Цифрлық бейнені сығу үшін компрессорды таңдаған кезде қалай сығуға тырысу керек?
9. Кодек деген не? Қандай кроссплатформалы кодектер бар?
10. Графикалық ақпаратты сығу алгоритмдері мен технологияларын атаңыз.
11. MPEG-сығу қандай идеяға негізделген? MPEG түрлерін атаңыз.
12. «Сығу алгоритмі» мен «формат» ұғымдарының айырмашылығын түсіндіріңіз. Жиі қолданылатын қандай форматтар бар?
13. Аудиокодек деген не?
14. Аудиоформаттардың қандай топтары бар?
15. iTunes Store-ден әуен тек қандай форматпен ғана жүктеледі?

ӨЗДІК ЖҰМЫС ТАПСЫРМАЛАРЫ

1. Графикалық ақпараттарды сығу деңгейлеріне мысалдар келтіріңіз:
а) сығу сапасын жоғалтпай сығу; б) сапасын жоғалтып сығу; в) адамның қабылдауы тұрғысынан сапасын жоғалтпай сығу;
г) табиғи түрде сапасын жоғалтып сығу; д) табиғи емес түрде сапасын жоғалтып сығу.
2. Графикалық ақпаратты сығу алгоритмдерін қолданатын жүйелерге мысалдар келтіріңіз: а) компрессия мен декомпрессия уақытына жоғары талап қоятын жүйелер; б) компрессия дәрежесіне және декомпрессия уақытына жоғары талап қоятын жүйелер;
в) компрессия дәрежесіне жоғары талап қоятын жүйелер.
3. Кескінді шығынсыз және шығынмен сығу алгоритмдеріне мысалдар келтіріңіз.
4. Цифрлық бейнені (видео) сығу технологияларының салыстырмалы сипаттамаларын көрсететін кесте құрыңыз.
5. Бейнені сығу стандарттарын атап, сипаттама беріңіз.
6. Бір бейнефайлды әртүрлі алгоритмдермен сығып, әртүрлі форматтарда сақтаңыз, нәтижелерін кестеде көрсетіңіз.
7. Графикалық ақпаратты (кескінді) сығу алгоритмдері мен технологияларының салыстырмалы сипаттамаларын, неге негізделгенін, қолданысын, артықшылықтары мен кемшіліктерін көрсететін кесте құрыңыз.
8. Бір түрлі-түсті суретті әртүрлі алгоритмдермен сығып, әртүрлі форматтарда сақтаңыз, нәтижелерін кестеде көрсетіңіз.
9. Бір ақ-қара түсті (сызба) суретті әртүрлі алгоритмдермен сығып, әртүрлі форматтарда сақтаңыз, нәтижелерін кестеде көрсетіңіз.
10. Бір түрлі-түсті фотоны әртүрлі алгоритмдермен сығып, әртүрлі форматтарда сақтаңыз, нәтижелерін кестеде көрсетіңіз. Тиімді сығу алгоритмі мен форматты анықтаңыз.
11. MPEG, JPEG форматтарына сипаттама беріңіз.
12. Танымал аудиоформаттардың салыстырмалы сипаттамаларын көрсететін кесте құрыңыз. Тиімдісін анықтаңыз.
13. Бір әуенді әртүрлі алгоритмдермен сығып, әртүрлі форматтарда сақтаңыз, нәтижелерін кестеде көрсетіңіз. Тиімдісін анықтаңыз.
14. MP3 форматының 4 режиміне сипаттама беріңіз.
15. Apple Inc жасаған ALAC кроссплатформалық аудиокодекке сипаттама беріңіз. AAC-пен, FLAC-пен салыстырыңыз.

ҚОРЫТЫНДЫ

Қазіргі заманғы ақпараттық технологиялардың мүмкіндіктерін, тиімді кодтау әдістерін қолдану арқылы ақпараттық жүйенің сақталатын және байланыс желісімен жіберілетін деректерінің көлемін, арнадағы жіберілу уақытын қысқартуға қол жеткізуге болады. Сонымен қатар, кодтау әдістері арна арқылы алмасатын ақпараттың шынайылығына әсерін тигізеді.

Заманауи ақпараттық жүйелердің ішкі жүйелері арасында мәтіндік, графикалық, дыбыстық, бейне файлдар түріндегі, форматтары әртүрлі, көлемі аса үлкен цифрланған ақпараттар ағыны алмасып жатады. Осы ақпараттар ағынының байланыс желісімен дұрыс жіберілуін қамтамасыз ету үшін кодтау әдістері қолданылады.

Кодтау әдістерінің қолданылу аясы кең, атап айтқанда:

– деректерді сақтаған немесе жіберген кезде олардың көлемдерін азайту мақсатымен деректерді сығу үшін;

– есептеу құрылғыларының жадтарындағы, цифрлік ленталар мен дисктердегі сақталынған деректерді қорғау үшін;

– желі аппаратурасының дұрыс жұмыс істемеулерінен және логикалық тізбектердегі шулардан қорғану үшін;

– бөгеуілге орнықты қатені түзететін кодтар шуы бар арналарда қабылданатын сигналдар әлсіз болғанда немесе жіберу қуаты шектеулі байланыс жүйелерінде хабарды сенімді жіберу үшін;

– әскери сала жүйелерінде қарсылас жақтың қасақана ұйымдастырылатын бөгеуілдерінен қорғану үшін;

– есептеулерде қателерді болдырмау маңызды болатын жүйелерде;

– көп қолданушысы бар, уақыт ішінде үлестірілген ақпараттық жүйелерде хабарды жіберу және жоғалған/бүлінген хабарды қалпына келтіру үшін тиімді қолданылады.

Цифрлық байланыс арналарындағы ақпараттың қауіпсіздігін, жіберу жылдамдығын арттыру бағытындағы ғылыми зерттеулер нәтижесінде жаңадан кодтау мен декодтаудың әдістері, кодтардың модификациялары, әртүрлі каскадтау әдістері ашылып, қолданысқа ие болды. Бұл кодтардың қатені анықтау мен түзету мүмкіндіктері артып, кеңейіп келеді.

Қазіргі заманғы байланыс жүйелері цифрлық түрдегі хабар алмасуды кеңінен қолданады. Мұндай цифрлық байланыс жүйелері үшін маңызды мәселелердің бірі – ол ақпаратты арнамен жібергенде жоғары сенімділікті, шынайлықты қамтамасыз ету болып табылады.

Байланыс арнасындағы болатын әртүрлі кедергілік әсерлер (шу, бөгеуіл) және хабарды қабылдау кезіндегі аппараттық істен шығулар цифрлық деректердің шынайлығын жоғалтуына, яғни бұрмалануына алып келеді.

Ақпараттың бұрмалануының салдары әртүрлі шығындарға, әсіресе әскери салаға, қауіпсіздікке, ғарыштық байланысқа қатысты жүйелерде апатты зардаптарға ұрындыруы мүмкін. Сондықтан да арнадағы шу деңгейінің жоғары жағдайында цифрлық байланыс жүйелеріндегі бөгеуілдермен күресудің ең тиімді құралы ретінде кодтау теориясының тиімді әдістері, бөгеуілге орнықты кодтау қолданылады. Мұндай бөгеуілге орнықты кодтар қабылданған хабардың қатесін анықтауға және түзетуге мүмкіндік береді.

Байланыс техникасының қарқынды дамуына байланысты тиімді декодерлер, Витерби алгоритмі, циклдік кодтар, жималаушы кодтар, Рид-Соломон коды, каскадтық кодтар және турбо кодтарға арналған алгоритмдер және көп шекті декодерлер кеңінен қолданыс енгізілді.

Алайда, шулы арналарда, соның ішінде ғарыш байланыс арналарында бөгеуілге орнықты алгоритмдерге қойылатын талаптар ұдайы артуда және де мұндай кодтардан арна энергетикасы бойынша үйлесімділікке жақын тиімділікпен декодтау, сонымен қатар қарапайым аппараттық жүзеге асыруды, жоғары энергетикалық ұтысты қамтамасыз ету талап етіледі.

ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР

1. Березюк Н.Т., Андрущенко А.Г. и др. Кодирование информации (двоичные коды). Изд-во «Вища школа», Харьков, 1978. - 252 с.
2. Блейхут Р. Теория и практика кодов, контролирующихся ошибки: Пер с англ./Под ред. К.Ш. Зигангирова. – М.: Мир, 1987
3. В.В.Лидовский. Основы теории информации и криптографии. <http://www.intuit.ru/studies/courses/2256/140/lecture/3914>.
4. Ватолин Д., Ратушняк А., Смирнов М., Юкин В. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео. - М.: ДИАЛОГ-МИФИ, 2002. - 384 с.
5. Вернер М. Основы кодирования. - Изд-во «Техносфера» Москва, 2004. - 288 с.
6. Гладких А.А. Теория помехоустойчивого кодирования. Методические указания. – Ульяновск: УлГТУ, 2005. - 34 с.
7. Золотарёв В.В., Овечкин Г. В. Помехоустойчивое кодирование. Методы и алгоритмы: Справочник /Под. ред. чл.-кор. РАН Ю. Б. Зубарева. – М.: Горячая линия–Телеком, 2004. - 126 с.: ил.
8. Иванюк А.А., Мусин С.Б. Специальные главы высшей математики: теория помехоустойчивого кодирования. Практикум. Белорусский государственный университет информатики и радиоэлектроники, 2007. - 33с.
9. Касами Т., Токура Н., Ивадари Ё. -Теория кодирования. 1978, 576 с.
10. Колесник В.Д. «Кодирование при передаче и хранении информации (алгебраич. теория блочных кодов)». М.: Высш. школа. 2009. - 550 с.
11. Королёв А.И. Коды и устройства помехоустойчивого кодирования. Научно-практическое издание. -Мн.: Бестпринт, 2002. –286 с.
12. Р. Морелос-Сарагоса. Искусство помехоустойчивого кодирования. Методы, алгоритмы, применение/ пер. с англ. В.Б.Афанасьева. - М.: Техносфера, 2006. – 320 с.
13. Першин В.Т. Основы современной радиоэлектроники. Учебное пособие - Ростов н/Д.: Феникс, 2009. - 541с.: ил. - (Высш. образов.).

14. Питерсон У., Уэлдон Э. Коды, исправляющие ошибки: Пер. с англ./Под ред. Р.Л. Добрушина и С.И. Самойленко. – М.: Мир, 1976
15. Ромашенко А., Румянцев А., Шень А. Заметки по теории кодирования. М.: МЦНМО, 2011. - 80с.
16. Сагалович Ю.Л. Введение в алгебраические коды / Учебное пособие. Изд. ИППИ РАН, 2010. - 302 с.
17. Семашко П.Г. Кодирование и декодирование линейных групповых кодов. Методическое пособие к лабораторной работе. – Минск: БГУИР, 2006. - 24 с.
18. Сидельников В.М. Теория кодирования. Изд: ФИЗМАТЛИТ, 2011. - 323 с.
19. Скляр Б. Цифровая связь. – М: “Вильямс”, 2003. – 1104 с.
20. Соловьева Ф.И. Введение в теорию кодирования: Учебное пособие / Новосибирский государственный университет. - Новосибирск, 2010. - 124 с.
21. Сэломон Д. Сжатие данных, изображений и звука. – М.: ТЕХНОСФЕРА, 2004.
22. Фрейман В.И. Теория электрической связи. Помехоустойчивое кодирование в телекоммуникационных системах уч.-метод. пособие. - Пермь: Изд. Перм. нац. исслед. политехн. ун-та, 2011. - 73 с.
23. Цымбал В.П. Теория информации и кодирования. Учебник. 4-е изд., перераб. и доп- Изд-во «Вища школа» Киев 1992. - 263 с.: ил.
24. Шульгин В.И. Основы теории связи. Часть 1. Теория и практика кодирования. / Учебное пособие – Харьков: Нац. аэрокосм.ун-т «Харьк. авиац. ин-т», 2005. - 196 с.
25. Шульгин В.И.. Основы теории передачи информации. Часть 2. Помехоустойчивое кодирование/ Учебное пособие. - Харьков: Нац. аэрокосм. ун-т « Харьковский авиацион. институт», 2003. - 87 с.