

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Інститут підготовки юридичних кадрів для СБУ  
Національного юридичного університету імені Ярослава Мудрого  
Харківський національний університет імені В. Н. Каразіна

**ОСНОВИ ОБРОБКИ ТА ЗАХИСТУ  
ВІДЕОІНФОРМАЦІЇ В  
ІНФОРМАЦІЙНИХ СИСТЕМАХ**

Навчальний посібник  
для курсантів та студентів вищих навчальних закладів

Харків  
2021

УДК 004.89(075.8)  
Б-24

*Затверджено до видання вченою радою Інституту підготовки юридичних кадрів для СБУ Національного юридичного університету імені Ярослава Мудрого (протокол №12 від 24.06.2021)*

**Автори:**

**В. В. Бараннік**, доктор технічних наук, професор; **Ю.М. Рябуха**, доктор технічних наук, старший дослідник **В. В. Твердохліб**, кандидат технічних наук.

**Рецензенти:**

**В. М. Безрук**, доктор техн. наук, професор (Харківський національний університет радіоелектроніки); **Ю. В. Стасєв**, доктор техн. наук, професор (Харківський національний університет Повітряних Сил ім. Івана Кожедуба).

**Бараннік В. В.**

Б-24 Основи обробки та захисту відеоінформації в інформаційних системах: навч. посіб. для студ. вищ. навч. закл. / В. В. Бараннік, Ю.М. Рябуха, В.В. Твердохліб. – Х. : ФОП Бровін О.В., 2021. –172 с. ISBN 978-617-8009-39-7

У навчальному посібнику систематизовано викладені сутність та основи обробки даних та захисту відеоінформаційних ресурсів, розглянуті математичні моделі та базові методи кодування відеоданих. Систематизовано викладені методологічні принципи побудови класифікаційних вимог до систем обробки та захисту відеоданих. Особливу увагу приділено розгляду методів та засобів формування, обробки і аналізу даних відеоінформаційного ресурсу. У систематизованому вигляді розглянуте комплексне питання щодо скорочення різних видів надмірності в рамках вимог відеосервесів. Всебічно викладено універсальні, статистичні та структурні підходи відносно побудови методів ефективного подання та захисту відеоданих.

Навчальний посібник призначений для курсантів та студентів вищих навчальних закладів, ад'юнктів, аспірантів, слухачів, науково-педагогічних та інженерно-технічних працівників.

ISBN 978-617-8009-39-7

УДК 004.89(075.8)

© Бараннік В. В., Рябуха Ю.М., Твердохліб В. В., 2021  
© Інститут підготовки юридичних кадрів для СБУ  
Національного юридичного університету імені Ярослава Мудрого, 2021

## З М І С Т

СПИСОК СКОРОЧЕНЬ.....	5
ВСТУП.....	6
Розділ 1. ОСНОВНІ ВИМОГИ ЩОДО ПОДАННЯ ДЛЯ ЗАХИСТУ ВІДЕОДАНИХ В АВТОМАТИЗОВАНИХ ТА ІНФОРМАЦІЙНИХ СИСТЕМАХ.....	9
1.1. Методологічні принципи побудови класифікаційних вимог до систем стиснення відеозображень.....	9
1.2. Аналіз вимог щодо компактного подання даних відеоінформаційного ресурсу в інформаційних системах .....	17
1.3. Об'єктивні та суб'єктивні міри оцінки спотворень відеоданих в процесі обробки.....	32
Контрольні питання та завдання	36
Розділ 2. МЕТОДИ ТРАНСФОРМУВАННЯ ВІДЕОЗОБРАЖЕНЬ В ТЕХНОЛОГІЯХ ЕФЕКТИВНОГО СИНТАКСИЧНОГО КОДУВАННЯ.....	38
2.1. Сутність та особливості трансформації відеоданих в теорії кодування.....	38
2.2. Синусоїдальні ортогональні перетворення.....	48
2.3. Несинусоїдальні ортогональні перетворення.....	55
2.4. Несинусоїдальні перетворення в базисі функцій Хаара.....	59
2.5. Базові принципи дискретного вейвлет-перетворення.....	63
Контрольні питання та завдання	71

Розділ 3. БАЗОВІ ТЕХНОЛОГІЇ ТА МЕТОДИ КОДУВАННЯ ВІДЕОІНФОРМАЦІЇ.....	73
3.1. Універсальні коди. Методи кодування за алфавітом джерела інформації.....	73
3.2. Методи кодування відеоданих з урахуванням виявлення ймовірно-статистичних закономірностей .....	83
3.3. Методи кодування з урахуванням виявлення структурних закономірностей у зображеннях.....	127
Контрольні питання та завдання	157
ЛІТЕРАТУРА.....	160

## СПИСОК СКОРОЧЕНЬ

ПР	–	повітряна розвідка
БПЛА	–	безпілотний літальний апарат
БЗП	–	бортовий запам'ятовувальний пристрій
ЗВІ	–	засоби відображення інформації
ПСВ	–	підсистема стиснення відеоданих
brb	–	bit per bit (біт на біт)
ОпП	–	оперативна пам'ять
ОП	–	ортогональні перетворення
ДКП	–	дискретне косинусне перетворення
ДПХ	–	дискретне перетворення Хаара
ДПУ	–	дискретне перетворення Уолша
ДВП	–	дискретне вейвлет-перетворення
RLE	–	Run Length Encoding
PSNR	–	peak-to-peak signal-to-noise ratio

## ВСТУП

Передові в економічному та військовому відношенні країни світу все більше уваги приділяють розробці наукоємних інформаційних технологій та систем. Отримують подальший розвиток основи теорії інформації та кодування. Особливого розвитку набувають сучасні технології та методи обробки відеоінформації та аудіоданих. Відеоінформаційні ресурси стають ключовою компонентою в процесі підтримки та прийняття рішень, в системах управління комплексами засобами автоматизації АСУ та інформаційним системам.

Навчальний посібник “Основи обробки та захисту відеоінформації в інформаційних системах” має сприяти підготовці спеціалістів в області інформаційних систем та технологій, комп’ютерних наук, кібербезпеки, телекомунікацій. Це зумовлено багатьма причинами, основними з яких є такі:

– сучасні інформаційні системи, які в тому числі використовуються в інтересах управління об’єктами критичної інфраструктури, потребують модернізації процесів обробки, передачі, прийому та аналізу відеоінформаційних ресурсів;

– у навчальних курсах інших університетів, де розглядається проблематика обробки кризисних відеоресурсів, недостатньо уваги приділяється питанням комплексної обробки відеозображень, дослідженню сучасних концепцій побудови методів кодування відеоданих. Запропонований навчальний посібник розкриває нові методи та засоби кодування, перетворення відеокадрів і призначається для навчання курсантів (студентів), слухачів, ад’юнктів (аспірантів), науково-педагогічних та інженерно-технічних працівників;

– у більшості видань, присвячених питанням цифрової обробки сигналів, окремо розглядаються питання стосовно ба-

зової та комплексної обробки, ці дослідження не систематизуються з позиції вимог користувачів відеоінформаційних сервісів. А даний навчальний посібник охоплює всі процеси отримання, кодування та перетворення відеоданих. Увага приділяється вивченню вимог щодо можливості здійснювати обробку цифрових відеозображень безпосередньо на засобах дистанційного мобільного моніторингу. Все це зумовило необхідність створення навчального посібника, в якому в систематизованому вигляді викладені теоретичні методи та засоби формування й обробки відеозображень в АСУ та інформаційних системах. Цей навчальний посібник має на меті надати знання про принцип дії, конструкцію, склад і методи розрахунків оцінки ефективності технологій та методів цифрової обробки відеозображень. Методи формування та обробки зображень розглядаються як в площині застосування для дистанційних засобів, так і для наземних засобів інформаційного забезпечення функціонування систем критичної інфраструктури.

Навчальний посібник необхідний для забезпечення лекційних, групових занять та самостійної роботи курсантів і студентів у галузі знань “Інформаційні системи” за навчальними дисциплінами “Основи обробки та захисту відеоінформації в автоматизованих та інформаційних системах”, “Інтелектуальні системи автоматизованої обробки інформації в АСУ” та при написанні курсових, дипломних і магістерських робіт зі спеціальності “Інформаційні системи та технології”.

За структурою навчальний посібник містить: анотацію, зміст, список скорочень, вступ, основний текст та літературу.

Основний текст підручника містить 3 розділи.

Надається інформацію щодо методологічних принципів побудови класифікаційних вимог систем обробки зображень.

Викладаються математичні моделі та базові методи кодування відеозображень. Вперше матеріал подається в систематизованому вигляді відносно комплексного питання щодо скорочення різних видів надмірності в рамках вимог відеосервісів. Розглядаються універсальні, статистичні та структурні підходи відносно побудови методів ефективного подання зображень.

Перший розділ надає інформацію щодо визначення основних вимог до подання відеоданих в автоматизованих та інформаційних системах. Розкриваються методологічні принципи побудови класифікаційних вимог до систем стиснення відеозображень. Визначаються об’єктивні та суб’єктивні міри оцінки спотворень відеоданих в процесі обробки.

У другому розділі викладені методи трансформування відеозображень в технологіях ефективного синтаксичного кодування. Розкривається сутність, класифікація та особливості трансформації відеоданих в теорії кодування. Досліджуються ортогональні перетворення та базові принципи дискретного вейвлет-перетворення.

У третьому розділі викладені базові технології та методи кодування відеоінформації. Розглядаються універсальні, статистичні та структурні підходи відносно побудови методів компактного подання відеозображень.

# Розділ 1

## ОСНОВНІ ВИМОГИ ЩОДО ПОДАННЯ ДЛЯ ЗАХИСТУ ВІДЕОДАНИХ В АВТОМАТИЗОВАНИХ ТА ІНФОРМАЦІЙНИХ СИСТЕМАХ

### 1.1. МЕТОДОЛОГІЧНІ ПРИНЦИПИ ПОБУДОВИ КЛАСИФІКАЦІЙНИХ ВИМОГ ДО СИСТЕМ СТИСНЕННЯ ВІДЕОЗОБРАЖЕНЬ

На даний час існує велика кількість принципів побудови та класифікації систем стиснення зображень. Але у разі обробки видових зображень повітряної розвідки (ПР) на борту авіаційних комплексів існують принципові особливості щодо їх обробки, які слід враховувати при побудові методологічних основ компресії. Забезпечення відеоінформацією з використанням ПР пов'язане з певними характерними умовами, а саме наявністю типових обмежень відносно [1; 2]:

1) вагово-габаритних характеристик безпілотних літальних апаратів (БПЛА). Більше 70 % від усієї стартової маси БПЛА приходить на двигун. Особливо з урахуванням тенденції, що намітилася за останні п'ять років, відносно мініатюризації БПЛА. У даний час створюються міні-БПЛА і мікро-БПЛА вагою, що не перевищує порядку 10 г, а розмірами, що не перевищують 15 мм;

2) потужності та щільності джерел живлення. Слід враховувати, що двигун споживає близько 90 % всієї потужності, а на забезпечення систем авіоніки та зв'язку приходить в се-

редньому 10 %. Отже, існує часове обмеження  $t_m$  на доставку даних з урахуванням енергетичних можливостей БПЛА;

3) характеристик продуктивності бортової апаратури обробки даних і неможливості інтегрування складних обчислювальних процесів на борту комплексу ПР;

4) часу  $t_{m,k}$  між формуванням сусідніх кадрів, що знаходиться в прямо пропорційній залежності від висоти та швидкості польоту бортового комплексу ПР, на якому здійснюється знімання інформації, тобто  $t_{m,k} \sim H$  і  $t_{m,k} \sim U_{\text{БПЛА}}$ .

Необхідно враховувати додаткові вимоги стосовно погодного режиму, що накладає обмеження на час польоту. Можливість ефективної видової ПР залежить від щільності опадів і температурного режиму (виконання завдання має здійснюватися при температурі від  $-40$  до  $+50$  °C).

Отже, бортовим комплексам ПР відповідають малі розміри антен і нестача енергетичних ресурсів на забезпечення бортових функцій обробки та передачі інформації. Можна виділити такі проблемні сторони організації передачі даних з бортових комплексів ПР [3].

По-перше, це нестача енергетичних можливостей для підтримки необхідної швидкодії при передачі зображень. Реальні швидкості передачі даних з борту БПЛА мають такі значення 10 Кбіт/с, 256 Кбіт/с, 656 Кбіт/с, 2 Мбіт/с. Для існуючих значень швидкості  $U_k$  передачі даних час  $T(W)_d$  доставки даних обсягом, який дорівнює  $W$  біт, з борту БПЛА визначається в основному часом  $T(W)_{\text{пер}}$  підготовки та передачі даних по каналу зв'язку і оцінюється за формулою

$$T(W)_d = T(W)_{\text{пер}} = W / U_k, \quad (1.1)$$

де  $U_k$  – швидкість передачі даних по каналу зв'язку з борту БПЛА.

Наслідком цього недоліку є:

а) несвоєчасна доставка оцифрованих зображень із заданою роздільною здатністю, тобто

$$T(W)_д > T(W)_{потр},$$

де  $T(W)_{потр}$  – потрібний час для доведення відеоданих необхідного обсягу, рівний  $W$  біт;

б) перевищення ліміту часу  $t_m$  доставки даних, що задається з урахуванням енергетичних обмежень бортової апаратури, тобто

$$T(W)_д > t_m;$$

в) значні затримки  $t_{м.к}$  доставки даних відносно часу, що задається з урахуванням обмежень між формуванням сусідніх кадрів, тобто

$$T(W)_д > t_{м.к}.$$

Слід зазначити, що використання всеспрямованого сигналу через його слабкість неможлива. Тому потрібні спрямовані наземні антени, щоб супроводжувати апарат по лінії візування. Але для міських умов, умов поганої видимості, гірської місцевості існують обмеження щодо прямої видимості. Отже, необхідні інші підходи, основними з яких є використання супутникових угруповань і систем зв'язку стільникової архітектури. Через значну вартість їх практичне використання дещо обмежене.

Фактори, що призводять до обмеження ефективності систем зв'язку БПЛА, наведені на рис. 1.1.

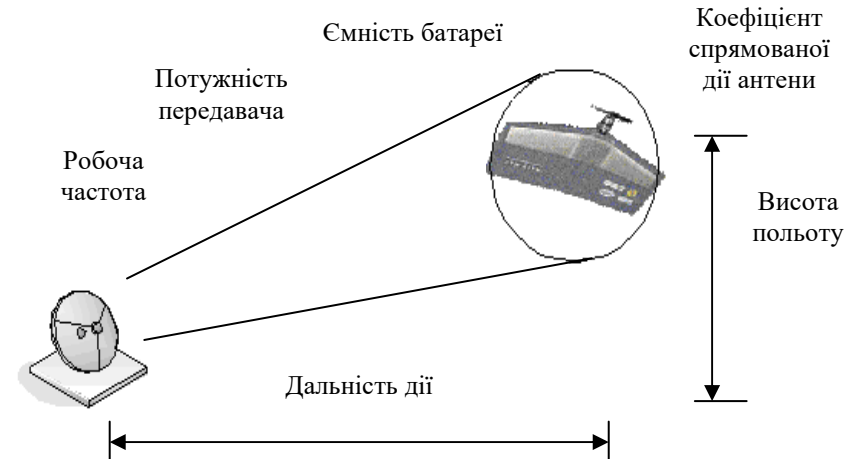


Рис. 1.1. Фактори ефективності системи зв'язку для бортових засобів ПР

Це викликає обмеження часу  $t_{с.з}$  сеансу зв'язку між БПЛА та одержувачем інформації. В умовах недостатньої видимості таке обмеження може додатково скоротити припустимий час на доставку даних з борту БПЛА. Тому процес організації повітряної розвідки з використанням БПЛА характеризується:

– з одного боку, сукупністю вимог споживачів розвіданих, а саме високою роздільною здатністю, наявністю декількох відео-фотокамер, необхідністю своєчасного доведення достовірної інформації аж до реального часу. Несвоєчасне доведення інформації призводить до її старіння та прийняття помилкових рішень;

– з іншого боку, відмінностями системи ПР на базі БПЛА у порівнянні з іншими засобами видової розвідки. Існують

умови, за яких необхідно здійснювати доведення інформації з використанням системи повітряної розвідки. Ці умови призводять до наявності обмежень, які впливають на процеси обробки даних на борту БПЛА і їхньої передачі в пункти призначення.

Отже, виникає *дисбаланс* між необхідними характеристиками процесів доставки даних, а саме часом обробки, передачі, якістю відновлюваних зображень, і реальними характеристиками для існуючих комплексів безпілотних авіаційних систем.

Розглянемо напрямки для зниження часу доведення інформації (рис. 1.2) [3].



Рис. 1.2. Напрямки підвищення оперативності доведення відеорозвідданих

Відповідно до виразу (1.1) на час доведення даних з бортового комплексу ПР впливають швидкість передачі даних по каналу зв'язку та обсяги переданих відеоданих.

Підвищення оперативності доведення інформації можливо на основі: 1) підвищення швидкості передачі даних з борту засобу ПР. Однак реалізації даного напрямку перешкоджають обмежені енергетичні можливості бортової апаратури передачі

даних; 2) упровадження інформаційно-обчислювальних систем, що мають високу швидкодію. Цей напрямок обмежується потужністю бортових джерел живлення та необхідністю додаткових фінансових витрат; 3) зменшення обсягів даних, що обробляються та передаються.

В обґрунтуванні третього напрямку необхідно зазначити, що видові зображення характеризуються наявністю різних класів надмірності (структурної, ймовірносно-статистичної, психовізуальної), кількість якої може досягати 95 %. Тому реалізація даного напрямку забезпечується за рахунок компактного подання відеоданих на бортовому комплексі ПР.

Системи, що забезпечують скорочення обсягів даних у цифровому вигляді при збереженні необхідної кількості інформації (на якісному та кількісному рівні), називаються системами підвищення інформативності або компактного подання даних. Компактне подання передбачає два етапи обробки: стиснення даних та їх відновлення (рис. 1.3). Використання технологій стиснення відеоданих забезпечить: зменшення цифрового обсягу відеоінформації; одержання кількісних класифікаторів окремих ознак зображень для їх дешифрування; підвищення якості відновленої відеоінформації, збільшення корисного часу для дешифрування; скорочення кількості помилок через емоційну реакцію оператора.

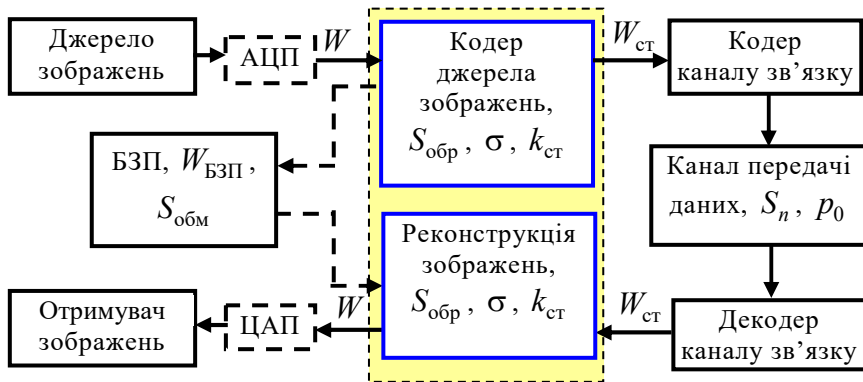


Рис. 1.3. Структурна схема інтегрування технології компресії в комплекс ПР

Роль підсистем стиснення відеоданих у процесі доставки розвідданих засобами повітряної розвідки з технічної сторони полягає: в скороченні обчислювального ресурсу; зниженні часових витрат на зчитування даних; збільшенні вільного місця на бортовому запам'ятовувальному пристрої (БЗП); зниженні вимог до характеристик телекомунікаційних технологій, у тому числі мереж доступу та транспортних мереж; скороченні часу передачі даних по каналах зв'язку; зниженні вимог до технічних параметрів каналів зв'язку; підвищенні якості відображуваної інформації; скороченні часу сприйняття інформація; скороченні енергетичних витрат на передачу стиснутих даних з борту.

Важливою ланкою в процесі прийняття рішення є оператор [3 – 5], тому особливе значення мають засоби відображення інформації (ЗВІ). У табл. 1.1 подано аналіз ролі ЗВІ та підсистем стиснення відеоданих (ПСВ) у процесі дешифрування видових зображень і прийняття рішень.

Таблиця 1.1

### Роль ПСВ у процесі відображення розвідувальної інформації

Завдання ЗВІ	Роль ПСВ
Відображення фрагментів інформаційних моделей загальної оперативної обстановки, сформульованих на екрані дисплея, з метою їх більш детального перегляду, коригування та наступного введення змін	1. Скорочення часу перегляду зображень. 2. Збільшення корисного часу на коригування
Візуальний контроль положення та динаміки переміщення об'єктів контролю в межах обраного географічного регіону	Підвищення оперативності контролю
Наочне відображення інформації про контрольовані процеси й об'єкти у реальному масштабі часу	Забезпечення доведення інформації в реальному масштабі часу
Дешифрування аерофотознімків, прив'язка до місцевості	Збільшення корисного часу на дешифрування

Характеристики методів стиснення залежать від вимог застосувань і умов їх інтегрування в загальний процес обробки зображень. Показники зниження обсягів зображень залежать від класу оброблюваних зображень, потужності обчислювальних засобів і можливості внесення спотворень у процесі їх обробки. Часові затримки на компресію (декомпресію) залежать від складності алгоритму; від платформи, на якій реалізований

алгоритм кодування (декодування); від виду реалізації, а саме програмної, апаратної та програмно-апаратної; від потужності обчислювальних комплексів. Перевага одного алгоритму перед іншим може дати можливість використання в обчисленнях алгоритму технологій нижнього рівня, типу ММХ, а також можливість розпаралелення етапів обробки.

## 1.2. АНАЛІЗ ВИМОГ ЩОДО КОМПАКТНОГО ПОДАННЯ ДАНИХ ВІДЕОІНФОРМАЦІЙНОГО РЕСУРСУ В ІНФОРМАЦІЙНИХ СИСТЕМАХ

Організація компактного подання зображень для бортових засобів ПР пов'язана з певними умовами.

*Перша умова* полягає в тому, що формовані бортовими системами цифрові зображення відрізняються високим ступенем насиченості дрібними об'єктами, контурами та перешкодами, що утворюються в процесі реєстрації. Для таких зображень відсутні області, що заповнені одним кольором, а області з плавним переходом кольорів мають обмежені розміри. Площа основного фону буде мінімальною, а концентрація дрібних об'єктів найвищою. Назвемо такі зображення зображеннями видової повітряної розвідки. Зображенням такого класу відповідають нестационарність імовірно-статистичних характеристик їх фрагментів, низька кореляція між елементами, концентрація високочастотних складових, велика ймовірність кольорних перепадів і значущих контурів. Це обмежує можливість використання методів стиснення, що базуються на виявленні ймовірно-статистичних закономірностей і обліку психовізуальних особливостей зорової системи.

*Друга умова* формується в результаті прояву в складній формі інформаційно-технологічного дисбалансу. Такий дисбаланс викликаний, з одного боку, відносно низькими швидкостями передачі даних з бортових засобів ПР і обмеженими обсягами бортових засобів зберігання даних, а з іншого – великими розмірами зображень видової повітряної розвідки. Дана обставина диктує вимогу для додаткового зниження обсягів зображень. Ступінь скорочення обсягів відеоданих оцінюється такими характеристиками [1; 6; 7].

1. Коефіцієнт стиснення  $k_{ст}$ , значення якого обчислюється за формулою

$$k_{ст} = W / W_{ст},$$

де  $W_{ст}$  – обсяг стиснутого зображення.

Фізичний зміст коефіцієнта стиснення має такі трактування:

- ступінь зменшення обсягу початкового зображення;
- середня кількість біт початкового подання, що приходить на один біт стиснутого опису.

Обсяг стиснутого зображення містить у собі витрати двійкових розрядів на подання:

- інформаційної частини – власне компактне (більш інформативне) подання початкового зображення;
- службової частини, що складається з відомостей про параметри та характеристики інформаційної моделі зображення, маркерних роздільників між кодовими конструкціями (у разі нерівномірного кодування), відомостей про початкове зображення та про версію програми-кодування, відомостей про структуру формату стиснутого зображення;

– коду програми компресії та декомпресії. Необхідність у таких даних відпадає, якщо на приймальному і передавальному боці відомі програми кодування й декодування.

Якщо витрати на службові дані не враховуються, така оцінка визначається як коефіцієнт стиснення  $k_{\text{инф}}$  по інформативних вибірках:

$$k_{\text{инф}} = W / W_{\text{инф}},$$

де  $W_{\text{инф}}$  – обсяг інформативної частини компактного подання зображення.

Такий показник використовується, щоб оцінити ефективність алгоритмів скорочення надмірності та ступінь впливу обсягів службових даних на загальний коефіцієнт стиснення.

Величина коефіцієнта стиснення більше одиниці означає наявність зменшення початкового обсягу. Навпаки, коли  $k_{\text{ст}} < 1$ , то маємо розширення початкового обсягу відеоданих.

При інтегруванні технологій компресії на бортових засобах ПР значення коефіцієнта стиснення має бути не менше заданого  $k_3$ , тобто повинна виконуватися нерівність  $k_3 \leq k_{\text{ст}}$ .

2. Показники обернені величині коефіцієнта стиснення:

а) середня кількість  $b_{\text{ср}}$  двійкових розрядів стиснутого подання, що приходяться на один елемент початкового подання:

$$b_{\text{ср}} = b / k_{\text{ст}},$$

де  $b$  – кількість розрядів на подання початкового елемента зображення.

Для повнокольорових зображень значення елемента дорівнює  $b = 24$  бітам, для кольорних складових багатоканального зображення під елементом видового зображення розуміється величина яскравості та  $b = 4; 8; 12$  бітам;

б) середня кількість біт  $\bar{b}_{\text{ср}}$  компактно поданого зображення, яка приходить на один біт початкового опису, що обчислюється за формулою

$$\bar{b}_{\text{ср}} = W_{\text{ст}} / W = W_{\text{ст}} / k_{\text{ст}} W_{\text{ст}} = 1 / k_{\text{ст}}. \quad (1.2)$$

У літературі величину  $\bar{b}_{\text{ср}}$  позначають як bpb (bit per bit – біт на біт).

Інтерес у такій оцінці полягає в тому, що величина  $\bar{b}_{\text{ср}}$  змінюється в межах від 0 до 1. Це дозволяє оцінювати ступінь відхилення показника  $\bar{b}_{\text{ср}}$  від крайніх границь. Рівність  $\bar{b}_{\text{ср}} = 0,7$  означає, що стиснуті відеодані становлять 70 % від початкового розміру. Якщо  $\bar{b}_{\text{ср}} > 1$ , то можна стверджувати, що обсяг відеоданих на виході кодера більше, ніж обсяг на вході, тобто спостерігається збільшення початкового обсягу зображення.

Похідним показником від  $\bar{b}_{\text{ср}}$  є величина

$$k = 100(1 - \bar{b}_{\text{ср}})\%.$$

Тут рівність  $k = 70\%$  означає, що на подання стиснутого зображення приходить на  $70\%$  менше двійкових розрядів, ніж на початкове подання;

в) середня кількість біт  $\bar{b}_e$  стиснутого подання, яка приходить на один елемент (піксел) початкового зображення, що оцінюється за формулою

$$\bar{b}_e = W_{\text{ст}} / Z_{\text{ряд}} Z_{\text{ств}}, \quad (1.3)$$

де  $Z_{\text{ряд}}$ ,  $Z_{\text{ств}}$  – відповідно кількість рядків і стовпців у кадрі зображення.

Величину  $\bar{b}_e$  зручно порівнювати зі значенням  $b$  до початку компресії. Зв'язок між величинами  $\bar{b}_{\text{ср}}$ ,  $\bar{b}_e$  і  $b$  на основі виразів (1.2), (1.3) і з урахуванням, що  $W = bZ_{\text{ряд}}Z_{\text{ств}}$ , буде мати такий вигляд:

$$\bar{b}_e = \bar{b}_{\text{ср}} W / Z_{\text{ряд}} Z_{\text{ств}} = \bar{b}_{\text{ср}} b Z_{\text{ряд}} Z_{\text{ств}} / Z_{\text{ряд}} Z_{\text{ств}} = \bar{b}_{\text{ср}} b = b / k_{\text{ст}}.$$

У літературі іноді величину  $\bar{b}_e$  позначають як  $\text{brp}$  (bit per pixel – біт на піксел).

3. Бітова швидкість  $r$  (bitrate). Даний показник є більш загальним у порівнянні з  $\text{brp}$  і  $\text{bpr}$ . Величина  $r$  у загальному випадку записується співвідношенням

$$r = \frac{f}{v} \left( \sum_{i=1}^v W(i)_{\text{ст}} + W_{\text{д}} \right),$$

де  $W(i)_{\text{ст}}$  – обсяг стиснутого  $i$ -го кадру ( $i$ -го фрагмента кадру зображення);

$f$  – частота кадрів (частота фрагментів усередині кадру), що створені джерелом відеоінформації в одиницю часу;

$v$  – кількість кадрів у складі відеопослідовності (кількість фрагментів усередині кадру);

$W_{\text{д}}$  – витрати кількості біт на подання коду програми розпакування (декодування).

Найкращою буде та технологія компресії, що забезпечує компактне подання відеоданих з найменшою бітовою швидкістю. При порівнянні методів стиснення може виявитися, що одні методи виграють за значенням коефіцієнтів стиснення в області їх високих значень, і навпаки, програють – в області низьких значень. Попереднє стиснення на борту дозволяє знизити обсяги переданих по каналах зв'язку відеоданих. Час  $T(W_{\text{ст}})_{\text{пер}}$  на передачу стиснутих даних по каналах зв'язку буде дорівнювати

$$T(W_{\text{ст}})_{\text{пер}} = \frac{W}{S_{\text{п}} k_{\text{ст}}} \text{ [біт/с]},$$

де  $S_{\text{п}}$  – швидкість передачі даних по каналах зв'язку.

**Третя умова** диктується економією енергетичного ресурсу бортового комплексу. Це стосується комплектації бортового засобу ПР низькопродуктивними обчислювальними системами (табл. 1.2, 1.3).

Таблиця 1.2

## Продуктивність деяких обчислювальних систем

Модель процесора	Тактова частота, МГц	Кількість операцій, млн опер./с			
		додавання/віднімання		множення/ділення	
		цілих	дійсних	цілих	дійсних
Системи, що вбудовуються					
DragonBall	66	60	6	7	3
StrongARM	200	200	100	20	8
Transmeta	600	500	250	60	25
VIA C3	800	800	350	80	40
Спеціалізовані системи					
Alpha 21064	166	166	58	16	5
Alpha 21168	266	233	110	24	10
Alpha 21264	500	450	260	55	26
Alpha 21364	1 000	900	530	290	190
IBM G5	500	500	260	50	24
Універсальні системи					
Intel Pentium	200	200	100	25	10
Intel Pentium II	600	600	300	150	70
Intel Pentium III	1 000	1 000	550	300	200
Intel Pentium IV	2 100	2 100	1 100	610	410

Отже, існують обмеження на технологічну та алгоритмічну реалізації методів обробки зображень, що інтегруються для бортової апаратури. Така вимога проявляється в накладенні обмежень на:

– кількість і тип обчислювальних операцій на стиснення (кодування)  $v_{ст}$  і відновлення (декодування)  $v_{в}$  зображення. Під типом обчислювальних операцій розуміються арифметичні (додавання, множення, ділення) і неарифметичні (порівняння, зчитування або запис на зовнішній запам'ятовуючий пристрій) операції. При цьому також ураховується кількість арифметичних операцій над цілими та дійсними числами;

– обсяг оперативної пам'яті та кількість обмінів з бортовим запам'ятовувальним пристроєм, які необхідні для здійснення обробки зображення;

– можливість реалізації алгоритмів компресії з використанням універсальних або спеціалізованих обчислювальних систем.

Часові витрати  $T_{ст.в}$  на стиснення та реконструкцію зображень оцінюються на основі використання виразу

$$T_{ст.в} = T_{ст} + T_{в},$$

де  $T_{ст}$ ,  $T_{в}$  – відповідно час на кодування та декодування відеоданих

$$T_{ст} = (v_{ст} / S_{обр}) + T_{обм}; \quad T_{в} = (v_{в} / S_{обр}) + T_{обм}.$$

Тут  $S_{обр}$  – швидкодія обчислювальної системи щодо виконання машинних операцій (швидкодія обробки);

$T_{обм}$  – сумарний час, що затрачується на обмін інформацією між оперативною пам'яттю (ОпП) і БЗП, необхідний для

забезпечення стиснення (відновлення) необхідного обсягу даних  $W$ ; визначається часом  $T_{\text{обм}}^{(1)}$  і кількістю таких обмінів  $v_{\text{обм}}$ :

$$T_{\text{обм}} = v_{\text{обм}} T_{\text{обм}}^{(1)} \approx v_{\text{обм}} (T_{\text{дст}} + T_{\text{зч/зп}} + T_{\text{ОпП-БЗП}}),$$

де  $T_{\text{обм}}^{(1)}$  – час, що витрачається на одноразовий обмін інформацією, обсягом  $w^{(1)}$  біт, між ОпП і БЗП. Затримка містить у собі: час доступу до необхідної ділянки диску  $T_{\text{дст}}$  (залежить від кількості обертів дисків БЗП за хвилину та часу підведення магнітних головок); час послідовного зчитування (запису) даних з (на) БЗП  $T_{\text{зч/зп}}$  (залежить від щільності та швидкості обертів дисків, а також обсягу одноразового зчитування/запису магнітними головками); час передачі даних між ОпП і БЗП  $T_{\text{ОпП-БЗП}}$  (залежить від швидкості  $S_{\text{обм}}$  передачі даних між ОпП і БЗП).

Основним параметром, що впливає на час виконання будь-якого перетворення, є продуктивність обчислювальної системи (тактова частота процесора)  $V_{\text{проц}}$ . Продуктивність обчислювальних систем подана в табл. 1.2. Одним із підходів для розрахунку продуктивності сучасних обчислювальних систем є пікова продуктивність комп'ютерів, що розраховується за формулою [1; 3]

$$V_{\text{проц}} = R_{\text{п}} = F_{\text{п}} n_{\text{п}} n_{\text{такт}} 10^{-6}, \quad (1.4)$$

де  $R_{\text{п}}$  – пікова продуктивність;  $F_{\text{п}}$  – частота процесора (ядра процесора), МГц;  $n_{\text{п}}$  – кількість процесорів або ядер у проце-

сорі;  $n_{\text{такт}}$  – кількість інструкцій (команд) процесора, що виконуються за один такт.

Пікова продуктивність вимірюється в терафлопсах (TFLOPS) і показує, скільки операцій за секунду виконує дана обчислювальна система (зазвичай маються на увазі операції над плаваючими числами розміру 64 біт у форматі IEEE 754):

$$\begin{aligned} 1 \text{ терафлопс} &= 1 \text{ трил. опер./с} = 1000 \text{ млрд опер./с;} \\ 1 \text{ TFLOPS} &= 1012 \text{ FLOPS} = 103 \text{ GFLOPS.} \end{aligned}$$

У табл. 1.3 наведена пікова продуктивність деяких сучасних комп'ютерів, що розрахована за допомогою виразу (1.4).

Таблиця 1.3

### Пікова продуктивність сучасних комп'ютерів

Тип процесора	Тактова частота $F_{\text{п}}$ , МГц	Кількість ядер $n_{\text{п}}$	Кількість інструкцій за такт $n_{\text{такт}}$	Пікова продуктивність $R_{\text{п}}$ , гигафлоп (млрд опер./с)
AMD Athlon 64	2 000	1	4	8
AMD Athlon 64 X2 4200	2 200	2	3	13,2
AMD Athlon II X4 645 (ADX645W)	3 100	4	3	37,2
AMD Phenom 9500 sAM2+	2 200	2	4	17,6

Закінчення табл. 1.3

Тип процесора	Тактова частота $F_{\Pi}$ , МГц	Кількість ядер $n_{\Pi}$	Кількість інструкцій за такт $n_{\text{такт}}$	Пікова продуктивність $R_{\Pi}$ , гигафлоп (млрд опер./с)
AMD Phenom II X6 1100T Black Edition (HDE00ZF)	4 000	4	4	64
Intel Pentium 4	3 000	1	2	6
Intel Core 2 Duo	2 400	2	4	19,2
Intel Core 2 Duo	3 000	2	4	24
Intel Core 2 Quad Q6600	2 400	4	4	38,4
Intel Core 2 Quad Q8300	2 500	4	4	40
Intel Core 2 Quad	3 500	4	4	56
Intel Core i 7-975 XE	3 300	4	4	52,8

Реальна продуктивність сучасних комп'ютерів  $R$  (на основі тесту LINPACK) становить порядку 80 – 95 % від пікової продуктивності  $R_{\Pi}$  та визначається виразом

$$V_{\text{проц}} = R \approx 0,8R_{\Pi} = 0,8(F_{\Pi} n_{\Pi} n_{\text{такт}} 10^{-6}).$$

Незважаючи на зазначену однозначність, у реальності пікова продуктивність, що розраховується у флопсах, не є ідеальною, оскільки неоднозначним є вже саме його визначення.

Крім того, величина флопс підпадає під вплив дуже багатьох факторів, прямо не пов'язаних із продуктивністю обчислювального модуля, таких, як пропускна здатність каналів зв'язку, продуктивність основної пам'яті й синхронність роботи кеш-пам'яті різних рівнів.

Тому продуктивність обчислювальної системи в дослідницьких цілях пропонується визначати через тактову частоту за формулою

$$V_{\text{проц}} = 0,8 \cdot F_{\Pi} 10^6 \text{ [тактів/с]}.$$

З урахуванням попередньої компресії зображень оперативність доставки розвідданих у системі ПР оцінюється сумарним часом  $T(W_{\text{ст}})_{\text{д}}$  на обробку та передачу стиснутих відеоданих з борту. Час доставки визначається в основному трьома базовими часовими складовими, а саме часовими витратами на компресію, передачу по каналах зв'язку та реконструкцію зображення. Така залежність задається формулою

$$T(W_{\text{ст}})_{\text{д}} = T_{\text{ст}} + T(W_{\text{ст}})_{\text{пер}} + T_{\text{в}},$$

де  $T(W_{\text{ст}})_{\text{д}}$  – час доведення по каналах зв'язку стиснутих відеоданих;  $T_{\text{ст}}$ ,  $T_{\text{в}}$  – відповідно часові витрати на стиснення і відновлення відеоданих.

Тут відповідно часові витрати на стиснення і відновлення визначаються за формулами

$$T_{\text{ст}} = v(Z_{\text{ряд}} Z_{\text{ств}})_{\text{ст}} / S_{\text{обр}}; \quad T_{\text{в}} = v(Z_{\text{ряд}} Z_{\text{ств}})_{\text{в}} / S_{\text{обр}},$$

де  $v(Z_{\text{ряд}}Z_{\text{ств}})_{\text{ст}}$ ,  $v(Z_{\text{ряд}}Z_{\text{ств}})_{\text{в}}$  – кількість операцій, що відводяться на стиснення та відновлення зображень, що містять  $Z_{\text{ряд}}Z_{\text{ств}}$  елементів.

Відповідно до обмеженої потужності джерел живлення на борту БПЛА існують обмеження на енергетичні витрати для реалізації процесів обробки та передачі даних  $P_{\text{пер}}$ . У той же час процеси компресії зображень на борту БПЛА пов'язані з необхідністю додаткових енергетичних витрат  $P(W)_{\text{ст}}$ . Однак за рахунок стиснення зображень забезпечується скорочення кількості розрядів, що передані з борту БПЛА. Це забезпечує зниження енергетичних витрат для апаратури передачі даних, тобто  $P(W_{\text{ст}})_{\text{пер}} < P_{\text{пер}}$ . Тому для характеристик технологій стиснення відеоданих на борту БПЛА повинна виконуватися умова  $P(W)_{\text{ст}} + P(W_{\text{ст}})_{\text{пер}} \leq P_{\text{пер}}$ , де  $P_{\text{пер}}$  – потужність, що відводиться для передачі даних з борту БПЛА (задана з урахуванням енергетичних можливостей бортової апаратури);  $P(W)_{\text{ст}}$  – потужність, необхідна для виконання процесів стиснення відеоданих обсягом  $W$  біт;  $P(W_{\text{ст}})_{\text{пер}}$  – потужність, необхідна для передачі стиснутих даних з борту БПЛА.

Вимога щодо зниження складності процесів обробки зображень диктується також необхідністю зменшення вартості бортової апаратури. Відомо, що саме витрати на бортові засоби зв'язку та авіоніки становлять до 90 % від сумарної вартості засобу ПР. Сумарна ж вартість сучасних засобів ПР може досягати порядку 10 млн долл. Тому економія засобів на бортову апаратуру дозволить істотно скоротити вартість усього комплексу ПР.

Слід зазначити, що вимога щодо зниження складності методів стиснення, тобто підвищення швидкості обробки в умовах збереження найбільшої інформативності зображень, знаходиться в обернено пропорційній залежності відносно коефіцієнта стиснення. Дійсно, чим менше часу приділяється на виявлення закономірностей і розподіл двійкових розрядів, тим менше буде ступінь стиснення. Багато методів припускають внесення спотворень в процесі компресії. У цьому випадку вдається зберегти необхідний рівень ступеня стиснення та підвищити швидкість обробки, але ціною тому є внесення більш істотних спотворень.

**Четверта умова** обумовлена тим, що повітряна розвідка є по суті справи дистанційним способом одержання відеоінформації про динамічно-мінливу обстановку і часом про заздалегідь невідомі об'єкти спостереження. Отже, слід урахувати, що:

– *по-перше*, фактично відсутні апріорні відомості про значеннєвий і синтаксичний зміст зображень ПР;

– *по-друге*, формуються найбільш достовірні та повні відеоінформаційні моделі про дистанційні об'єкти спостереження (розвідки);

– *по-третє*, найчастіше одержуване зображення піддається автоматичній обробці без участі людини. У цьому напрямку розробляються методи та технології автоматичного дешифрування зображень ПР;

– *по-четверте*, у процесі дешифрування зображень виникає необхідність їх масштабування. Дана вимога має на увазі легкість зміни розмірів зображення до розмірів вікна активного застосування. Тут одні алгоритми дозволяють легко масштабувати зображення прямо під час декомпресії, інші ж не

тільки не дозволяють легко масштабувати, але й збільшують імовірність появи завадових артефактів після застосування стандартних алгоритмів масштабування до декомпресованого зображення. Наприклад, такий недолік властивий для методів на JPEG-платформі при обробці зображень ПР. Характер внесених алгоритмом JPEG перекручувань такий, що зменшення або збільшення зображення може призвести до спотворень або втрати дрібних об'єктів;

– *по-п'яте*, отримані зображення ПР піддаються багаторазовому аналізу, в процесі якого здійснюється повторна обробка (редагування). Тут важливо забезпечити мінімальний ступінь погіршення якості зображення при його повторному збереженні після обробки. Слід відзначити, що багато методів із втратою інформації можуть істотно спотворити зображення за кілька ітерацій обробки.

Наслідком розглянутих особливостей є вимоги відносно:

а) інтеграції в систему компресії зображень ПР тих компонентів, ефективність яких не залежить від наявності апріорної інформації;

б) забезпечення мінімізації рівня спотворень, внесених у процесі обробки зображень.

Треба відзначити, що, дотепер не вироблено загальної моделі надійного визначення якості зображень, із позиції їх сприйняття зоровою системою людини та ймовірності розпізнавання в автоматичному режимі. Тому існує багато методик оцінки ступеня внесених спотворень у процесі обробки зображень.

Спотворення якості відновлених зображень можуть проявлятися у вигляді:

– розмиття дрібних об'єктів аж до їх повного зникнення;

– появи блочності (при загрубленні низькочастотних складових);

– ефекту “снігу” (різка зміна кольору в окремих точках);

– ефекту Гіббса, тобто утворення своєрідного “німба” (ореола) навколо контурів із різкими переходами кольорів (викликається втратами у високочастотних складових);

– муарів (розмиття чітких ліній);

– слабких смуг за напрямком розгорнення.

### 1.3. ОБ'ЄКТИВНІ ТА СУБ'ЄКТИВНІ МІРИ ОЦІНКИ СПОТВОРЕНЬ ВІДЕОДАНИХ В ПРОЦЕСІ ОБРОБКИ

Для оцінки рівня спотворень, що внесені у процесі стиснення, використовуються:

а) *суб'єктивні міри перекручування* – визначаються на основі візуальних порівнянь;

б) *об'єктивні міри перекручування* – можуть бути обчислені кодером/декодером без особистої участі людини.

**Суб'єктивні міри.** Найкраще втрати якості оцінює людське око. Стиснення зображення можна вважати відмінним, якщо на око неможливо відрізнити оригінал від стиснутого зображення.

Кращий спосіб вимірювання якості відеоінформації ґрунтується на методі експертних оцінок.

Метод експертних оцінок містить у собі такі етапи:

– за результатами виконання тестових завдань визначається група експертів;

– кожний експерт переглядає початкову та декодовану відео послідовність і виставляє оцінку за деякою шкалою;

– аналізуються оцінки всіх експертів.

**Об’єктивні міри.** Ступінь перекручувань при обробці виводних зображень оцінюється такими кількісними показниками.

1. Середньоквадратичний показник  $\sigma$  погрішності значень пікселів відновленого зображення відносно початкового:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{Z_{\text{ряд}}} \sum_{j=1}^{Z_{\text{ств}}} (a_{ij} - a'_{ij})^2}{Z_{\text{ряд}} Z_{\text{ств}}}}.$$

Тут два зображення, одне з яких  $a_{ij}$ ,  $a'_{ij}$  – відповідно  $(i; j)$ -те значення початкового та реконструйованого зображення на приймальній стороні.

Згідно з цим критерієм зображення буде значно зіпсовано при зміні яскравості лише на 5%. У той же час зображення зі “снігом” (різкими стрибками значень кольорів окремих точок) будуть трактуватися як такі, що “майже не змінилися”.

2. Іншим критерієм є максимальне відхилення  $\delta$  від оригіналу:

$$\delta = \max_{i,j} |a_{ij} - a'_{ij}|.$$

Дана міра вкрай чутлива до сплеску окремих пікселів, тобто в зображенні може змінитися тільки один піксел, і за даним критерієм визнається, що зображення буде сильно зіпсованим.

3. Найбільш популярною мірою оцінки якості зображення є показник  $h_{\text{ст}}$  пікового співвідношення сигнал/шум (PSNR – peak-to-peak signal-to-noise ratio).

$$h_{\text{ст}} = 20 \lg \left( \frac{255}{\sqrt{\sum_{i=1}^{Z_{\text{ряд}}} \sum_{j=1}^{Z_{\text{ств}}} (a_{ij} - a'_{ij})^2 / Z_{\text{ряд}} Z_{\text{ств}}}} \right),$$

де  $a_{ij}$ ,  $a'_{ij}$  – відповідно початкове та відновлене значення  $(i; j)$ -го елемента;  $Z_{\text{ряд}}$ ,  $Z_{\text{ств}}$  – відповідно кількість рядків і стовпців у кадрі зображення.

Ця міра аналогічна середньоквадратичному відхиленню, але користуватися нею зручніше через логарифмічний масштаб шкали.

З урахуванням обмежень на ступінь втрат якості відновлених зображень має виконуватися нерівність  $h_{\text{ст}} \geq h_{\text{необ}}$ , де  $h_{\text{необ}}$  – необхідне значення пікового співвідношення сигнал/шум.

До інших найбільш вживаних критеріїв оцінки якості зображення відносяться [3; 6; 7]:

– коефіцієнт крос-кореляції  $K$ :

$$K = \frac{\sum_{i=1}^{Z_{\text{ряд}}} \sum_{j=1}^{Z_{\text{ств}}} [a_{ij} \cdot a'_{ij}]}{\sum_{i=1}^{Z_{\text{ряд}}} \sum_{j=1}^{Z_{\text{ств}}} [a_{ij}]^2};$$

– вірність зображення (image fidelity)  $AD$ , що оцінюється за формулою

$$AD = \frac{\sum_{i=1}^{Z_{\text{ряд}}} \sum_{j=1}^{Z_{\text{ств}}} [a_{ij} - a'_{ij}]^2}{\sum_{i=1}^{Z_{\text{ряд}}} \sum_{j=1}^{Z_{\text{ств}}} [a_{ij}]^2}.$$

Залежно від ступеня та характеру внесених перекручувань розрізняють три рівні їх градації, а саме:

а) найвищий рівень якості реконструйованих зображень – без втрати інформації. Даний рівень відповідає повній відсутності спотворень, тобто забезпечується відповідність зображень біт у біт;

б) із втратами якості відтворених зображень – відповідає випадку, коли в процесі обробки похибки вносилися, але вони не позначаються на їх візуальному сприйнятті;

в) найменший рівень – із втратами інформації. Такий рівень відповідає ситуації, коли внесені в процесі стиснення та відновлення перекручування значно впливають на візуальне сприйняття одержуваних зображень.

Проблематичність забезпечення високої якості декомпресованих зображень полягає в наявності обернено-пропорційної

залежності від ступеня стиснення та складності обробки. Іншими словами, існує протиріччя між вимогою забезпечення високої якості зображень і підвищенням ступеня стиснення в умовах заданої складності алгоритму компресії або між зниженням складності обробки в умовах заданої продуктивності обчислювальної системи та ступеня стиснення.

**П'ята умова** диктується тим, що:

– відеодані передаються по бездротових каналах зв'язку, що характеризується наявністю більш складної заводової обстановки порівняно з провідними каналами;

– відсутня можливість повторної передачі пакетів даних у разі їх втрати. Втрати пакетів відбуваються як під впливом помилок, так і в результаті переповнення буферних пристроїв відправника (бортового засобу ПР) і одержувача інформації.

Тому потрібно забезпечувати стійкість стиснутих відеоданих до помилок у каналі зв'язку та втрат пакетів у процесі їх передачі. Дана вимога означає локальність порушень у зображенні при псуванні або втраті фрагмента переданого файла.

Вимога стійкості до помилок і втрат пакетів у процесі передачі по каналах зв'язку також суперечить забезпеченню високого ступеня стиснення та зниженню складності обробки. Це пов'язане з уведенням додаткових надлишкових розрядів для забезпечення корекції помилок та зі збільшенням часу обробки для проведення заводостійкого кодування та збільшення обсягу стиснутих відеоданих.

### **Контрольні питання та завдання**

1. Назвіть шляхи підвищення оперативності доведення інформації.

2. Назвіть основні показники якості систем стиснення відеозображень.

3. Які вимоги висуваються відносно систем стиснення для комплексів повітряної розвідки?

4. Які існують класи надмірності відео зображень?

5. Які етапи обробки включає процес компактного подання відеоданих?

6. Яке призначення мають системи підвищення інформативності?

7. Яка роль підсистем стиснення відеоданих у процесі доставки відеоданих?

8. Яка роль підсистем стиснення відеоданих у процесі відображення відеоінформації?

9. Назвіть вимоги щодо компактного подання даних відеоінформаційного ресурсу в інформаційних системах.

10. Який фізичний зміст коефіцієнта стиснення?

11. Назвіть показники обернені величині коефіцієнта стиснення.

12. Який фізичний зміст показника бітової швидкості?

13. У якому вигляді можуть проявлятися спотворення якості відновлених зображень?

14. Які міри використовуються для оцінки рівня спотворень, що вносяться у процесі стиснення?

15. Дайте визначення суб'єктивної міри перекручування.

16. Дайте визначення об'єктивної міри перекручування.

17. Які етапи містить включає метод експертних оцінок?

18. Якими кількісними показниками оцінюється ступінь перекручувань при обробці відеозображень?

19. Який фізичний зміст показника пікового співвідношення сигнал/шум (PSNR)?

## Розділ 2

# МЕТОДИ ТРАНСФОРМУВАННЯ ВІДЕОЗОБРАЖЕНЬ В ТЕХНОЛОГІЯХ ЕФЕКТИВНОГО СИНТАКСИЧНОГО КОДУВАННЯ

### 2.1. СУТНІСТЬ ТА ОСОБЛИВОСТІ ТРАНСФОРМАЦІЇ ВІДЕОДАНИХ В ТЕОРІЇ КОДУВАННЯ

У результаті виконання ортогональних перетворень утворюються масиви коефіцієнтів (компонент) перетворень. Надалі масиви коефіцієнтів (компонент) перетворень називатимемо *трансформантами*, а сам процес перетворення *трансформацією*.

Під *трансформованим зображенням* розуміється зображення, подане коефіцієнтами перетворень [2; 8; 9].

У зв'язку з переходом на цифрові системи обробки інформації, далі розглядаються тільки дискретні ортогональні перетворення.

Оцифроване зображення може піддаватися перетворенню поблочно або повністю; в цьому випадку відповідно одержуємо поблочно трансформоване або трансформоване в цілому зображення.

На даний час в системах стиснення зображень використовуються одновимірні, двовимірні і тривимірні ортогональні перетворення (ОП). Відповідно існують одноетапні, двоетапні і трьохетапні схеми виконання багатовимірних ортогональних перетворень.

На етапі такої передоброби безпосереднього зменшення надмірності не відбувається. Проте формовані трансформанти мають порівняно з початковим поданням зображення певні переваги [2; 10], обумовлені такими інтерпретаціями перетворень:

– двовимірне перетворення можна розглядати як розкладання зображення в узагальнений двовимірний спектр, а коефіцієнти трансформанти – як амплітуди відповідних спектральних складових. Іншими словами, в результаті ортогонального перетворення здійснюється переведення початкового сигналу з просторово-часової області в спектрально-частотну область.

Спектр – це величини коефіцієнтів, з якими відповідні спектральні складові підсумовуються, що в результаті і дає сигнал. Окремі спектральні складові, на які розкладається сигнал, часто називають базисними функціями;

– компоненти трансформанти можна розглядати як вагові коефіцієнти, за якими необхідно підсумувати базисні зображення, щоб одержати початкове зображення. Дана інтерпретація дозволяє розглядати процес перетворення як процес апроксимації зображення відповідними базисними функціями;

– двовимірне перетворення можна розглядати як такий поворот  $n^2$ -вимірної системи координат, в якій перетворюване зображення, що складається з  $n^2$  елементів, представляється  $n^2$ -вимірним вектором, при якому кореляція між його компонентами (компонентами трансформанти) зводиться до мінімуму.

За рахунок властивостей перетворень може досягатися:

- а) концентрація енергії початкового фрагмента зображення у відносно малій кількості компонент трансформанти;
- б) декореляція між компонентами трансформанти.

Дані особливості трансформант перетворень дозволяють:  
– передавати меншою кількістю двійкових розрядів або відкидати частину компонент трансформанти;

– кодувати компоненти трансформанти на основі статистичних методів;

– виявляти ланцюжки компонент з нульовими значеннями.

Передача з внесенням похибок допускається для компонент високочастотних спектральних складових. Це пояснюється тим, що зоровий апарат людини менш чутливий до зміни високочастотних складових зображень (дрібні деталі) в порівнянні зі зміною великих деталей [1; 11; 12].

Концентрація енергії у відносно малій кількості компонент трансформанти і декореляція між ними призводять до збільшення ступеня нерівномірності розподілу ймовірності появи значень компонент. Це створює можливість для застосування ймовірнісно-статистичних кодів.

Зображення мають двовимірну структуру, що пояснює наявність структурних і статистичних залежностей по двох напрямках фрагментів зображень.

Для передоброби в системах стиснення зображень найбільшого поширення набули два способи виконання двовимірних ортогональних перетворень [10; 13].

Перший спосіб полягає в розгортці **двовимірного масиву в одновимірну послідовність**. Двовимірний масив відліків зображення за допомогою різних видів розгортки перетворюється в одновимірну послідовність  $X(n)$ . Одержана послідовність піддається одновимірному ортогональному перетворенню. Пряме і обернене перетворення для такого способу обчислюється відповідно за формулами

$$C(n) = H(n)X(n),$$

$$X(n) = H^T(n)C(n),$$

де  $X(n)$  – вектор-стовпець відліків зображення;

$H(n)$  – матриця дискретних значень базисних функцій ортогонального перетворення ( $n = \log_2 N$ );

$H^T(n)$  – транспонована матриця дискретних значень базисних функцій;

$C(n)$  – вектор коефіцієнтів перетворення (трансформанта);

$N$  – кількість елементів вектора відліків.

Основними *недоліками* описаного способу є:

– неможливість отримання й аналізу двовимірного спектра;

– можливість обліку і аналізу тільки одновимірних кореляційних взаємозв'язків уздовж розгортки.

Інший спосіб виконання двовимірного ортогонального перетворення заснований на використанні властивості роздільності ортогональних базисів. Ця властивість полягає в можливості використання для виконання двовимірних ортогональних перетворень одновимірних ортогональних базисів. У цьому випадку процедура прямого і оберненого ортогонального перетворення двовимірного масиву розмірності  $(N_1 \times N_2)$  у матричному вигляді описується такими виразами:

$$C(k, \ell) = F(k)X(i, j)_{k, \ell} F^T(\ell);$$

$$X(i, j) = F^T(k)C(k, \ell)_{i, j} F(\ell),$$
(2.1)

де  $X(i, j)_{k, \ell}$  – масив відеоданих, утворений на базі однієї з площин кольорорізнисної моделі зображення;

$k, \ell$  – відповідно індекс рядка і стовпця елемента масиву  $X(i, j)_{k, \ell}$ ,  $k = \overline{1, N_1}$ ;  $\ell = \overline{1, N_2}$ ;

$C(k, \ell)$  – матриця компонент трансформанти перетворення масиву відеоданих;

$F(k), F^T(\ell)$  – відповідно вектор дискретних значень базисних функцій ОП і його транспонований вигляд.

З аналізу даних виразів виходить, що пряме і обернене перетворення виконується в два етапи (двопрохідна схема).

Двопрохідна реалізація двовимірного ОП, що задається виразом (2.1), передбачає:

1) виконання одновимірного дискретного перетворення стовпців масиву відеоданих та формування проміжного масиву  $H(k, \ell)$ :

$$H(k, \ell) = F(k)X(i, j)_{k, \ell}, \quad (2.2)$$

де  $H(k, \ell)$  – проміжний масив розмірності  $(N_1 \times N_2)$ ;

2) виконання одновимірного ОП для рядків проміжного масиву  $H(k, \ell)$ :

$$C(k, \ell) = H(k, \ell)F^T(\ell). \quad (2.3)$$

Співвідношення (2.2) та (2.3) дозволяють сформулювати трансформоване подання  $C(k, \ell)$  зображень на основі двопрхідної схеми виконання ортогонального перетворення.

У загальному випадку двохпрохідна реалізація оберненого двовимірного ОП у матричному вигляді задається такими співвідношеннями:

1) на першому проході виконується одновимірне обернене ОП перетворення стовпців трансформанти  $C(k, \ell)^*$  та формується проміжний масив  $H(i, j)$ :

$$H(i, j) = F(i)C(k, \ell)_{i,j},$$

де  $H(i, j)$  – проміжний масив розмірності  $(N_1 \times N_2)$ ,

$F(i)$  – вектор дискретних значень базисних функцій ОП;

2) другий прохід пов'язаний з виконанням одновимірного оберненого ОП для рядків проміжного масиву  $H(i, j)$ :

$$X(i, j)^* = H(i, j)F^T(j),$$

де  $X(i, j)^*$  – масив відеоданих після етапу оберненого ОП;

$F^T(j)$  – транспонований вектор дискретних значень базисних функцій ОП.

Базисні функції  $F(k)$  ортогонального перетворення за визначенням мають відповідати вимогам ортогональності і ортонормованості. Відповідно безліч ортогональних і ортонормованих функцій утворює ортогональний базис.

Наведені вище двовимірні перетворення і обернені їм перетворення можна обчислити в результаті  $N_1 N_2$ -кратного застосування алгоритмів, використовуваних для обчислення відповідних одновимірних перетворень.

Основними принципами побудови ОП є:

1) наявність або відсутність залежності формування базисних функцій від елементів початкових відеопослідовностей, що апроксимуються;

2) тип базисних функцій ОП, на основі яких здійснюється апроксимація (перетворення) відеопослідовностей.

Тип базисної функції визначається:

– видом функціональної залежності, яка лежить в основі побудови базисів ОП;

– наявністю комплексної складової;

– чутливістю компонент трансформант ОП.

Відповідно до першого принципу ОП поділяють на перетворення:

а) із залежними від послідовностей, що апроксимуються, базисними функціями. У цьому випадку базисні функції формуються для кожної оброблюваної відеопослідовності, що, з одного боку, дозволяє мінімізувати помилку апроксимації, а з іншого, вимагає наявності могутніх обчислювальних засобів. Вимога щодо підвищення продуктивності диктується значною складністю таких перетворень. У свою чергу висока складність апроксимації обумовлена знаходженням коваріаційних моментів різних порядків, що потребує багатократного обходу по всіх елементах оброблюваної послідовності. Для таких перетворень кількість операцій буде значно відхилена від лінійної залежності (порядку  $N^4 - N^6$ ). На виконання таких перетворень потрібно витратити до 90 % часу від усього часу на стиснення зображень. До перетворень даного класу відносяться перетворення по власних кореляційних моментах, відоме також як перетворення Карунена – Лоева, і розкладання по сингулярних значеннях.

На даний час існує декілька підходів щодо скорочення кількості операцій на перетворення Карунезна – Лоева. Проте вони як правило пов'язані зі збільшенням обсягів оперативної пам'яті і ступеня спотворень. Це обмежує використання даного класу ОП для обробки видових зображень на бортових комплексах повітряної розвідки;

б) з незалежними від початкових даних базисними функціями. Базисні функції вважаються наперед відомими для кодера і декодера. Цей принцип дозволяє скоротити витрати кількості операцій на передобробку зображень. Основними прикладами такого класу ОП є дискретне косинусне перетворення (ДКП), дискретне синусне перетворення, дискретне перетворення Хаара (ДПХ) і варіації дискретних перетворень Уолша (ДПУ).

Другий принцип дозволяє додатково поділити ОП залежно від типу функцій, які лежать в основі побудови їх базисів. Даний принцип застосовується для ОП із заданими базисними функціями. Для побудови базисів таких ОП використовуються:

– гармонічні базисні функції, наприклад, косинусні і синусні;

– несинусоїдальні ортогональні функції, до яких відносяться: функції Радемахера, функції Хаару і функції Уолша. Ці ортогональні функції складаються з квадратних або прямокутних хвиль. Тому їх ще називають шматково-постійними (шматково-лінійними) функціями. На основі даних функцій будуються базиси перетворень Хаару, Уолта – Адамара і модифікованого перетворення Уолта – Адамара.

Синусоїдальні (гармонічні) ортогональні функції розділяються на комплексно-дійсні і дійсні. Прикладом перетворення на основі комплексно-дійсних базисів є перетворення

Фур'є і його модифікації. Ортогональними перетвореннями, в яких використовується дійсний базис, є дискретно-косинусне і дискретно-синусне перетворення.

Розглянемо такий показник класифікації ортогональних перетворень, як чутливість компонент їх трансформант.

Під **чутливістю компонент трансформант** (коефіцієнтів ОП) розуміється наявність для них залежності від елементів перетворюваних послідовностей. Залежно від кількості оброблених елементів, що впливають на значення компонент трансформант, виділяють такі типи чутливості:

– ортогональні перетворення, коефіцієнти яких мають глобальну чутливість;

– ортогональні перетворення, трансформанти яких містять компоненти з глобальною і локальною чутливістю.

У разі глобальної чутливості кожен коефіцієнт перетворення є функцією всіх координат простору вхідних послідовностей. Прикладами ОП, для яких усі коефіцієнти мають властивість глобальної чутливості, є ДКП і ДПУ.

Під **локальною чутливістю** розуміється залежність коефіцієнтів ОП від частини елементів перетворюваних послідовностей. Наприклад, для ДПХ коефіцієнти перетворення мають як глобальну, так і локальну чутливість.

З позиції використання ОП для передобробки в системах стиснення зображень важливими показниками ортогональних перетворень є також:

1. Середньоквадратична помилка апроксимації, що оцінюється показником середньоквадратичної похибки:

$$\sigma = \sqrt{\frac{\sum_{i=0}^{N_1-1} \sum_{j=0}^{N_2-1} (x_{i,j}^{(B)} - x_{i,j}^{(n)})^2}{N_1 N_2}},$$

де  $x_{i,j}^{(n)}$  – елемент початкового зображення;

$x_{i,j}^{(B)}$  – елемент відновленого зображення;

$N_1 \times N_2$  – розмірність зображення по вертикалі і горизонталі відповідно.

Даний показник залежить від типу ортогонального базису і способу його виконання.

2. Динамічний діапазон компонент трансформант  $B_{\text{тр}}$  розраховується за формулою

$$B_{\text{тр}} = |y_{\max}| - |y_{\min}|;$$

$$y_{\max} = \max_{\substack{0 \leq k \leq N_1-1; \\ 0 \leq \ell \leq N_2-1}} (y(k; \ell)); \quad y_{\min} = \min_{\substack{0 \leq k \leq N_1-1; \\ 0 \leq \ell \leq N_2-1}} (y(k; \ell)).$$

Тут  $y_{\max}$  і  $y_{\min}$  – відповідно максимальне і мінімальне значення компоненти в трансформанті ортогонального перетворення.

3. Наявність цілочисельних базисних функцій.

4. Кількість операцій множення і додавання, що витрачаються на виконання ОП.

## 2.2. СИНУСОЇДАЛЬНІ ОРТОГОНАЛЬНІ ПЕРЕТВОРЕННЯ

Найбільш поширеним ОП у класі базисів на синусоїдальних функціях для систем стиснення є дискретне косинусне перетворення. Таке перетворення використовується в методах на JPEG-платформі.

На відміну від перетворення Фур'є, яке застосовує для розкладання сигналу синусні і косинусні частотні складові, в ДКП використовуються тільки косинусні складові.

Вибір ДКП як перетворення для систем стиснення обґрунтовується такими причинами:

1) дискретне косинусне перетворення серед інших ортогональних перетворень є найбільш близьким до перетворення Карунена – Лоєва у плані декореляції (для більшості сигналів);

2) ДКП відповідає перетворенню Фур'є, згідно з якими усуваються розриви на краях послідовності і ряд Фур'є сходиться набагато швидше;

3) стиснення на основі ДКП забезпечує кращі характеристики при обробці оптичних зображень, текстурованих ділянок зображень, плавних переходів, сигналів гауссівської природи. У цьому випадку майже вся енергія початкових елементів зображення концентрується в 25 % компонент трансформанти ДКП;

4) ДКП асимптотично оптимальне для всіх марковських сигналів кінцевого порядку і базисні функції добре апроксимують власні вектори тепліцевих матриць;

5) ДКП забезпечує невелике значення середньоквадратичного відхилення відновлених зображень, за рахунок чого можлива побудова ефективних алгоритмів фільтрації і стиснення

зображення. Процеси стиснення зображень на основі ДКП і на основі вейвлет-перетворення забезпечують близькі в статистичному плані значення по коефіцієнту стиснення і по піковому відношенню сигнал/шум. При зменшенні розмірів масивів відеоданих (наприклад, у разі потреби скоротити час компресії і декомпресії), для яких виконується апроксимація, вираш виявляється на стороні ДКП обробки.

Гармонічний аналіз не має просторової локалізації, оскільки не мають просторової локалізації гармонічні функції. Тому перед виконанням перетворення (трансформації) колірна площина зображення поділяється на масиви розміром  $N_1 \times N_2$  [13]. Поділ на масиви здійснюється для зниження кількості операцій на обробку та підвищення ступеня когерентності (корельованості) оброблюваних даних.

Двопрохідна реалізація двовимірного ДКП полягає в тому, що:

1) на першому проході виконується одновимірне дискретне косинусне перетворення стовпців масиву відеоданих. Формується проміжний масив  $H(k, \ell)$ :

$$H(k, \ell) = F(k)X(i, j)_{k, \ell}, \quad (2.4)$$

де  $H(k, \ell)$  – проміжний масив розмірності  $N \times N$ .

З урахуванням системи виразів для вектора базисних функцій  $F(k)$  –

$$F(k) = \begin{cases} \frac{1}{\sqrt{N}} & \rightarrow k = 1; \\ \sqrt{\frac{2}{N}} \cos \frac{(2N+1)k\pi}{2N} & \rightarrow k = \overline{2, N}. \end{cases}$$

співвідношення (1.8) набуде вигляду

$$\begin{aligned} H(1, \ell) &= \frac{1}{\sqrt{N}} \sum_{k=1}^N X_{k\ell}^{(i, j)}; \\ H(k, \ell) &= \sqrt{\frac{2}{N}} \sum_{k=1}^N X_{k\ell}^{(i, j)} \cos \frac{(2N+1)k\pi}{2N}, \quad i = \overline{2, N}, \end{aligned} \quad (2.5)$$

де  $X_{k\ell}^{(i, j)}$  – елемент масиву  $X(i, j)$ , розташований на перетині  $k$ -го рядка і  $\ell$ -го стовпця масиву відеоданих;

2) другий прохід пов'язаний з виконанням одновимірного ДКП для рядків проміжного масиву  $H(k, \ell)$ :

$$C(k, \ell)_{\text{ДКП}} = H(k, \ell)F^T(\ell). \quad (2.6)$$

Співвідношення (2.5) та (2.6) дозволяють сформулювати трансформоване подання  $C(k, \ell)$  зображень на основі двопрхідної схеми виконання ортогонального перетворення.

Матриця  $F(3)$  базисних (косинусних) коефіцієнтів для перетворення розміром  $8 \times 8$ , має вигляд

$$F_{\text{ДКП}}(k; \ell) = \begin{bmatrix} 0,354 & 0,354 & 0,354 & 0,354 & 0,354 & 0,354 & 0,354 & 0,354 \\ 0,490 & 0,416 & 0,278 & 0,098 & -0,098 & -0,278 & -0,416 & -0,490 \\ 0,462 & 0,191 & -0,191 & -0,462 & -0,462 & -0,191 & 0,191 & 0,462 \\ 0,416 & -0,098 & -0,490 & -0,278 & 0,278 & 0,490 & 0,098 & -0,416 \\ 0,354 & -0,354 & -0,354 & -0,354 & 0,354 & -0,354 & -0,354 & 0,354 \\ 0,278 & -0,490 & 0,098 & 0,416 & -0,416 & -0,098 & 0,490 & -0,278 \\ 0,191 & 0,462 & 0,462 & -0,191 & -0,191 & 0,462 & -0,462 & 0,191 \\ 0,098 & 0,278 & 0,416 & -0,490 & 0,490 & -0,416 & 0,278 & -0,098 \end{bmatrix}.$$

У результаті застосування до блока зображення розміром  $8 \times 8$  пікселів дискретного косинусного перетворення одержимо двовимірний спектр, що також має розмір  $8 \times 8$  відліків. Іншими словами, 64 числа, будучи відліками зображення, перетворюються в 64 числа, що являтимуться відліками його ДКП-спектра. Для  $8 \times 8$  ДКП система базисних функцій задається формулою

$$F_{\text{ДКП}}(k; \ell) = \cos\left[\frac{(2k+1)u\pi}{16}\right] \cos\left[\frac{(2\ell+1)v\pi}{16}\right],$$

а самі базисні функції виглядають подібно наведеними на рис. 2.1.

Можна відмітити, що чим нижче і правіше в матриці ДКП його компонента, тим більш високочастотним деталям зображення вона відповідає.

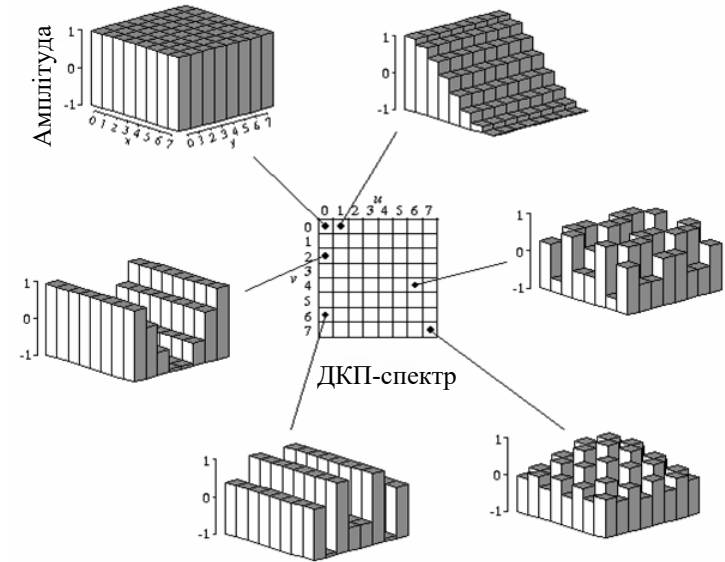


Рис. 2.1. Базисні функції для ДКП розміром  $8 \times 8$

Трансформанта дискретного косинусного перетворення має такі особливості [1; 14; 15]:

1) значення компоненти у верхньому лівому кутку трансформанти з координатами (1; 1) пропорційне середній яскравості локального фрагмента зображення;

2) компоненти лівої верхньої області трансформанти характеризують ступінь насиченості блока зображення низькочастотними перепадами. До низькочастотних перепадів відносять плавні і ступінчасті зміни рівня яскравості або координати кольору. Оскільки дані особливості є переважаючими, то основна енергія (компоненти з найбільшими значеннями) сконцентрована саме в цій області трансформанти;

3) значення компонент у нижній правій області трансформанти залежать від ступеня насиченості блока зображення

дрібними деталями й імпульсними сплесками. Для середньо-насичених реалістичних зображень високочастотні перепади не є характерними. Отже, в даній області трансформанти містяться менша кількість енергії початкового сигналу, а відповідно компоненти мають найменші значення;

4) значення компонент ДКП зменшуються за діагональним зигзагом зліва-направо, зверху-вниз;

5) дискретне косинусне перетворення використовує укрупнену по всьому блоку статистику даних, а значить й інтегровану взаємну кореляцію, величина якої буде майже 0,9. Це дозволяє для широкого класу зображень одержувати трансформанти ДКП з такими властивостями: компоненти трансформанти з великими значеннями сконцентровані відносно малої області трансформанти, а компоненти з мінімальними значеннями займають велику площу трансформанти;

б) оскільки статистичні характеристики відеоданих відрізняються від гауссівських процесів, то між компонентами трансформант ДКП існує статистична залежність;

7) двійкові подання компонент трансформант містять зони нульових елементів, причому розміщення і довжина нульових зон залежить від координати відповідної компоненти в трансформанті. Для компонент, які відповідають низькочастотним складовим, нульові зони розміщуються у середині і в кінці двійкового подання. Для високочастотних же компонент характерно розміщення нульових зон на початку двійкового подання.

Двопрохідна реалізація оберненого двовимірного ДКП задається такими співвідношеннями:

1) на першому проході виконується одновимірне обернене дискретне косинусне перетворення стовпців трансформанти

$C(k, \ell)_{\text{ДКП}}^{\bullet}$ , у результаті чого формується проміжний масив  $H(i, j)$ :

$$H(i, j) = F(i)C(k, \ell)_{i, j}, \quad (2.7)$$

де  $H(i, j)$  – проміжний масив розмірності  $N \times N$ .

Вираз (2.7) набуде вигляду

$$H(1, j) = \frac{1}{\sqrt{N}} \sum_{i=1}^N C_{i, j}^{(k, \ell)};$$

$$H(i, j) = \sqrt{\frac{2}{N}} \sum_{i=1}^N C_{i, j}^{(k, \ell)} \cos \frac{(2N+1)i\pi}{2N}, \quad i = \overline{2, N},$$

де  $C_{i, j}^{(k, \ell)}$  – елемент масиву  $C(k, \ell)_{\text{ДКП}}^{\bullet}$ , розташований на перетині  $i$ -го рядка і  $j$ -го стовпця трансформанти;

2) другий прохід пов'язаний з виконанням одновимірного оберненого ДКП для рядків проміжного масиву  $H(i, j)$ :

$$X(i, j)^{\bullet} = H(i, j)F^T(j),$$

де  $X(i, j)^{\bullet}$  – масив відеоданих після етапу оберненого ДКП, що належить одній з площин кольорорізнисної моделі зображення,  $X(i, j)^{\bullet} = \{x_{ij}^{\bullet}\}$ ;

$F^T(j)$  – транспонований вектор дискретних значень базисних функцій ДКП.

До недоліків дискретно-косинусного перетворення слід віднести таке:

- динамічний діапазон трансформанти перевищує динамічний діапазон відліків зображення у декілька разів. Це призводить до збільшення обсягу пам'яті, який займає зображення при переході до спектрального подання (у вигляді трансформант перетворення), за допомогою виконання ДКП;

- усі коефіцієнти ДКП мають глобальну чутливість і відповідну роздільну здатність тільки в частотній області. Для обчислення коефіцієнта перетворення, що має глобальну чутливість, необхідно виконати більше арифметичних операцій, ніж для коефіцієнта, що має локальну чутливість;

- усі арифметичні операції ДКП є дійсними.

Зазначені недоліки підтверджують складність технічної реалізації ДКП та значні часові витрати на їх виконання і це обмежує застосування при обробці зображень у реальному масштабі часу.

### 2.3. НЕСИНУСОЇДАЛЬНІ ОРТОГОНАЛЬНІ ПЕРЕТВОРЕННЯ

У 1923 р. Уолш одержав повну систему ортонормованих прямокутних функцій, яка доповнює систему функції Радемахера і відома тепер як система функцій Уолша. Безліч функцій Уолша зазвичай поділяється на три групи, в залежності від порядку розташування окремих функцій. Загальноприйнятими вважаються такі впорядкування: впорядкування за частотою (за Уолшем); діадичне впорядкування (за Пелі); природне впорядкування (за Адамаром). Кожна з вказаних систем має додаткові властивості. Наприклад, енергетичний спектр перетворення

Уолша з упорядкуванням за Уолшем є інваріантним відносно діадичного зсуву початкових відліків. Енергетичний спектр перетворення Уолша з упорядкуванням за Адамаром – інваріантний відносно циклічних зсувів початкових відліків.

Повну систему функцій Уолша, визначену на одиничному інтервалі  $[0,1)$ , можна поділити на дві групи парних і непарних функцій відносно точки  $t = 0,5$ . Ці парні і непарні функції аналогічні синусам і косинусам відповідно і тому їх позначають як  $sal$  (синусоподібні функції Уолша) і  $cal$  (косинусоподібні функції Уолша). Більш докладно математичний розгляд системи  $sal-cal$  функцій проведено Піхлером [13].

Базис перетворення Уолша будується на лінійно-постійних функціях і має глобальну чутливість.

Дискретизація функцій Уолша ( $walw(n; t)$ ), зображених на рис. 2.2, а, у восьми рівнорозташованих точках формує матрицю  $(8 \times 8)$ , показану на рис. 2.2, б. У загальному випадку виходить матриця  $(N \times N)$ . Такі матриці позначатимемо  $H(n)_w$ , де  $n = \log_2 N$ .

Припустимо, що  $u_i$  і  $v_i$  – цифри  $i$ -го розряду в двійковому поданні цілих чисел  $u$  та  $v$  відповідно, тобто

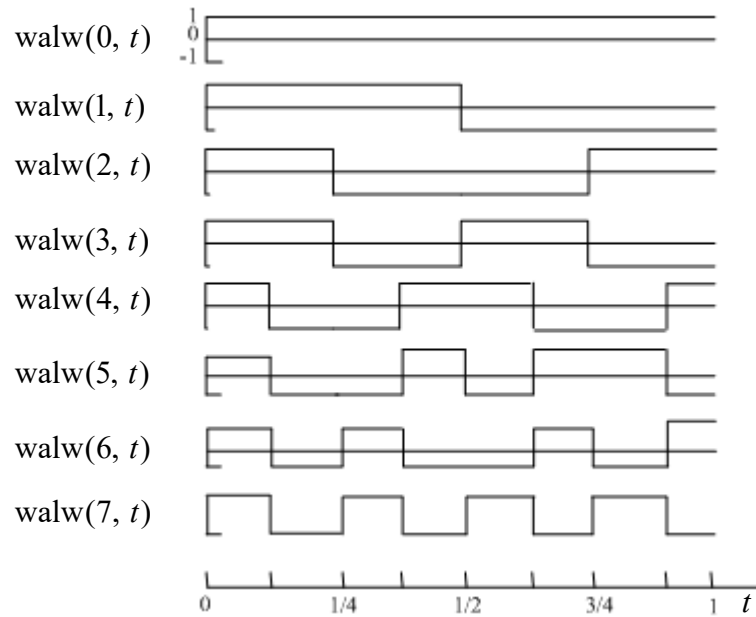
$$u_{10} = (u_{n-1}u_{n-2} \dots u_1u_0)_2, \quad v_{10} = (v_{n-1}v_{n-2} \dots v_1v_0)_2.$$

Тоді елементи  $h_{u,v}^w$  матриці  $H(n)_w$  матимуть такий вигляд:

$$h_{u,v}^w = (-1)^{\sum_{i=0}^{n-1} r(u)_i v_i}; \quad u, v = \overline{0, N-1},$$

$$r(u)_0 = u_{n-1}; \quad r(u)_1 = u_{n-1} + u_{n-2}; \quad r(u)_2 = u_{n-2} + u_{n-3}; \dots;$$

$$r(u)_{n-1} = u_1 + u_0.$$



$$H_w(3) = \begin{matrix} & a & & & & & & & \\ \left[ \begin{array}{cccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{array} \right] & \begin{array}{l} \leftarrow \text{walw}(0, t) \\ \leftarrow \text{sal}(1, t) \\ \leftarrow \text{cal}(1, t) \\ \leftarrow \text{sal}(2, t) \\ \leftarrow \text{cal}(2, t) \\ \leftarrow \text{sal}(3, t) \\ \leftarrow \text{cal}(3, t) \\ \leftarrow \text{sal}(4, t) \end{array} \\ & б & & & & & & & \end{matrix}$$

Рис. 2.2. Функції Уолша, впорядковані по Уолшу, при  $N = 8$ :  
 $a$  – неперервні;  $б$  – дискретні

Перші вісім дискретних функцій, упорядкованих за Уолшем, мають вигляд, наведений на рис. 2.2, б.

Пряме ДПУ визначається виразом

$$y(u; v) = \frac{1}{N} \sum_{i=1}^N h_{u,i}^{(w)} x(i; j); \quad u, v = \overline{0, N-1}.$$

Обернене ДПУ визначається формулою

$$x(i; j) = \frac{1}{N} \sum_{u=1}^N y(u; i) h_{u,i}^{(w)}.$$

Перевагами дискретного перетворення Уолша слід вважати:

- цілочисельність дискретних значень базисних функцій, що визначає простоту технічної реалізації і високу обчислювальну ефективність дискретного перетворення Уолша;

- інваріантність енергетичного спектра до різних видів зсувів;

- динамічний діапазон трансформант Уолта, який відповідає динамічному діапазону відліків зображення.

Тому ДПУ набуло широкого застосування в цифровій обробці сигналів завдяки простій технічній реалізації, яка забезпечується видом базисних функцій. Вони мають прямокутну форму і набувають лише двох значень: “1” і “-1”. Це дозволяє виключити в процесі виконання ДПУ операції множення. Дані властивості перетворення Уолша підкреслюють їх переваги для використання в системах стиснення зображень, інтегрованих для бортових систем як повітряного, так і космічного базування.

Разом з тим ДПУ властиві *недоліки*, а саме:

– дискретне перетворення Уолша у разі його використання для трансформації зображень з плавним переходом кольорів дає більший рівень спотворень порівняно з дискретним косинусним перетворенням;

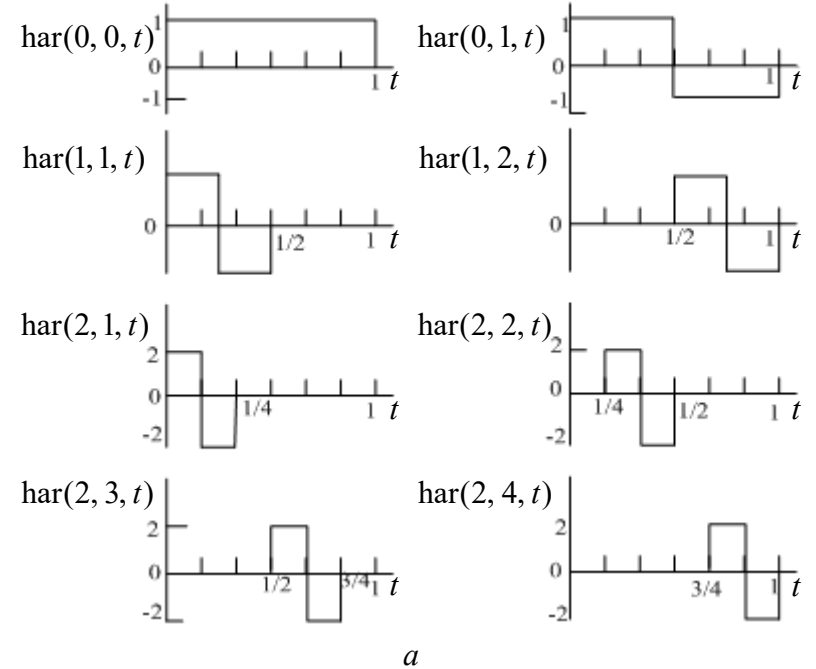
– так само як для ДКП усі коефіцієнти ДПУ мають глобальну чутливість і відповідно роздільну здатність лише в частотній області.

## 2.4. НЕСИНУСОЇДАЛЬНІ ПЕРЕТВОРЕННЯ В БАЗИСІ ФУНКЦІЙ ХААРА

Безліч функцій Хаара ( $\text{har}(n; m; t)$ ), які утворюють періодичну, ортонормовану і повну системи функцій [13], було запропоновано ним у 1910 р. На рис. 2.3, *a* зображені перші вісім функцій Хаара. Рекурентне співвідношення, що дозволяє одержати  $\text{har}(n; m; t)$ , має вигляд  $\text{har}(0; 0; t) = 1, t \in [0, 1)$ :

$$\text{har}(r; m; t) = \begin{cases} 2^{r/2}, & \rightarrow \frac{m-1}{2^r} \leq t < \frac{m-\frac{1}{2}}{2^r}; \\ -2^{r/2}, & \rightarrow \frac{m-\frac{1}{2}}{2^r} \leq t < \frac{m}{2^r}; \\ 0, & \rightarrow \text{для решти } t \in [0, 1); \end{cases} \quad (2.8)$$

$$0 \leq r \leq \log_2 N \quad \text{і} \quad 1 \leq m \leq 2^r.$$



$$H(3)_h = \begin{matrix} a \\ \left[ \begin{array}{cccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{array} \right] \begin{matrix} \} N/N \\ \} N/N \\ \} N/4 \\ \} N/2 \end{matrix} \end{matrix}$$

*b*

Рис. 2.3. Функції Хаара для  $N=8$ :  
*a* – неперервні; *b* – дискретні

Дискретизація системи функцій Хаара, показаних на рис. 2.3, а, формує матрицю, зображену на рис. 2.3, б, кожен рядок якої є дискретною функцією Хаара  $\text{har}(n; m; t)$ . Одержані таким чином матриці використовуються для перетворення Хаара і позначаються  $H(n)_h$ , де  $n = \log_2 N$ .

Коефіцієнти перетворення Хаара  $Y(k)_x$ , де  $k = \overline{0, N-1}$ , які відповідають вхідній послідовності  $X(m)$ ,  $m = \overline{0, N-1}$ , формуються у результаті обчислення прямого перетворення

$$Y(n)_x = \frac{1}{N} H(n)_h X(n),$$

де  $H(n)_h$  – матриця Хаара розміром  $N \times N$ .

Матриця  $H(n)_h$  формується у результаті дискретизації безлічі функцій Хаара  $\{\text{har}(r; m; t)\}$ , визначених виразом (2.8). Матриця Хаара для  $N = 8$  записується у вигляді, наведеному на рис. 2.3, б.

Аналіз матриці  $H(n)_h$  показує, що:

1)  $N/2$  коефіцієнтів перетворення Хаара відповідають кореляції двох сусідніх точок у просторі вхідних послідовностей  $N/4$  коефіцієнтів, які відповідають зв'язкам чотирьох сусідніх точок і т. д. до  $N/N$  коефіцієнтів, які у свою чергу відповідають усім  $N$  координатам простору вхідних послідовностей. Це означає, що область трансформації у разі перетворення Хаара має властивість як локальної, так і глобальної чутливості. Властивість глобальної чутливості для перетворення Хаара відноситься до перших двох коефіцієнтів;

2) одна частина функцій набуває цілочисельних значень, а інша – дійсних значень. Дійсні значення мають вигляд  $k\sqrt{2}$  ( $k = 1, 2, 3, \dots$ ). Така властивість на відміну від перетворення ДКП дозволяє використовувати цілочисельний ненормований алгоритм виконання ДПХ, дискретні значення функції якого набуватимуть значень 0 та  $\pm 2^r$ .

Слід зазначити, що трансформанти перетворення Хаара характеризуються значною структурною надмірністю, яка полягає в наявності великої кількості коефіцієнтів, значення яких прямує до нуля. Зазначена властивість дозволяє на основі перетворення Хаара будувати ефективні алгоритми стиснення зображень.

До переваг дискретного перетворення Хаара слід віднести такі:

– трансформанта перетворення Хаара містить компоненти, що мають як глобальну, так і локальну чутливість. Цим і забезпечується роздільна здатність коефіцієнтів Хаара одночасно в частотній і в часовій областях;

– простота технічної реалізації й висока обчислювальна ефективність дискретного перетворення Хаара. Так, відповідно до алгоритму Ендрюса для здійснення перетворення Хаара потрібно  $2(N-1)$  операцій додавання/віднімання і  $N$  операцій множення. За допомогою алгоритму Кулі – Тьюки для виконання ДПХ потрібно виконати  $\log_2 N$  операцій двійкових інверсій,  $2(N-1)$  операцій додавання/віднімання і  $N$  операцій множення;

– для деяких класів зображень ДПХ забезпечує менше значення середньоквадратичного показника похибки ніж при використанні трансформації ДПУ;

- динамічний діапазон трансформант Хаара відповідає динамічному діапазону відліків зображення;
- “гладку” структуру трансформант.

## 2.5. БАЗОВІ ПРИНЦИПИ ДИСКРЕТНОГО ВЕЙВЛЕТ-ПЕРЕТВОРЕННЯ

Дискретне вейвлет-перетворення (ДВП) – реалізація вейвлет-перетворення з використанням дискретного набору масштабів і перенесень вейвлету, заснованого на використанні рекурентних співвідношень для обчислення найбільш точних вибірок заданої функції материнського вейвлету з подвоєним розрізненням при переході до наступного рівня (масштабу). Це перетворення розкладає значення елементів початкового зображення на взаємно ортогональний набір вейвлетів і утворює вектор тієї ж довжини, що і вхідний. У вихідному векторі багато елементів мають значення менші, ніж у вхідному, аж до нульового рівня. Це відповідає факту, що він розкладається на набір вейвлетів (функцій), які ортогональні до їх паралельного перенесення та масштабування, тобто елементи початкового зображення подаються тією ж самою або меншою кількістю коефіцієнтів вейвлет-спектра. Це свідчить про те, що вейвлет-перетворення можна використовувати для попередньої обробки зображень перед їх стисненням. Дискретне вейвлет-перетворення у частотному плані розкладає початкове зображення на високочастотні  $y_H(n)$  і низькочастотні  $y_L(n)$  області, обробляючи кожен рядок і стовпець початкового зображення за допомогою частотних фільтрів, значення коефіцієнтів яких ( $h_0, g_0$ ) безпосередньо залежать від значень масштабуючої функції материнського вейвлету, формуючи на виході чотири

основні квадратури  $LL, LH, HL$  та  $HH$  (рис. 2.4), які містять інформацію про значення чотирьох різних частотних діапазонів [16; 17].

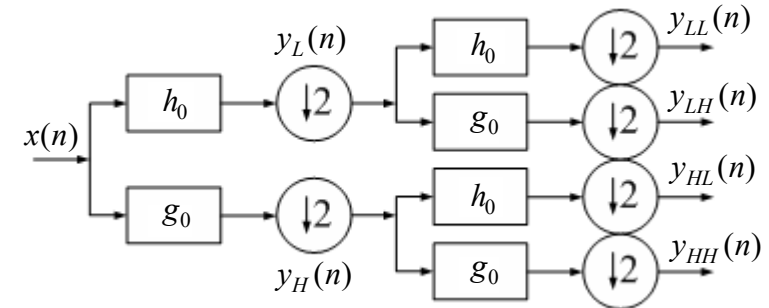


Рис. 2.4. Формування квадратури вейвлет-перетворення

Кількість елементів у кожній області можна скоротити вдвічі, користуючись теоремою Котельникова, тому що частотна область відносно вихідного зображення щораз при проходженні через фільтр зменшується вдвічі. Для усунення зайвих елементів використовується функція проріджування відліків.

З погляду часової області ДВП апроксимує початкове зображення за допомогою масштабуючих вейвлет-функцій, формуючи на виході набір коефіцієнтів, серед яких виділяють укрупнені масштабуючі та уточнюючі деталізуючі.

Вейвлет-перетворення має практично повний обсяг інформації про значення елементів вихідного зображення та разом із цим володіє надмірністю, виходячи з властивостей укрупнених масштабуючих і уточнюючих деталізуючих коефіцієнтів. Для відновлення елементів початкового зображення достатньо знати значення коефіцієнтів вейвлет-перетворення.

Ідея полягає в тому, щоб масштабувати вейвлет у кратну постійну кількість разів, і зміщувати його в часі на фіксовану відстань, яка залежить від масштабу. При цьому всі зсуви одного масштабу повинні бути попарно ортогональні – такі вейвлети називаються ортогональними. При такому перетворенні виконується згортка сигналу з деякою функцією (так званою скейлінг-функцією) і з вейвлетом, пов'язаним із цією скейлінг-функцією. У результаті утворюється “згладжена” версія початкового зображення та набір “деталей” (рис. 2.5 і рис. 2.6, складові  $L$  і  $H$  трансформант відповідно), що відрізняють згладжене зображення від оригіналу.

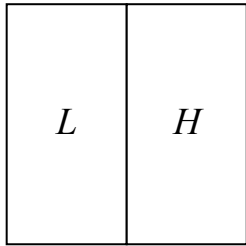


Рис. 2.5. Одержання квадратур вейвлет-перетворення при обробці зображень по рядках

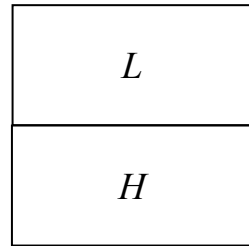


Рис. 2.6. Одержання квадратур вейвлет-перетворення при обробці зображень по стовпцях

Послідовно застосовуючи таке перетворення до рядків і стовпців початкового зображення, можна отримати результат потрібного ступеня детальності (гладкості) і набір елементів, що деталізують (квадратури  $LL$ ,  $LH$ ,  $HL$  і  $HH$ ). Так само, застосовуючи масштабування материнського вейвлету кілька разів до  $LL$ -квадратури та виділяючи з неї так само набори масштабуючих і уточнюючих коефіцієнтів, можна одержати кратного рівня вкладені квадратури.

Вейвлет-функції дозволяють проводити аналіз сигналів, які характеризуються високочастотним компонентом протягом короткого проміжку часу та низькочастотними коливаннями при розгляді великих часових масштабів. Стандартні віконні перетворення дозволяють проаналізувати або високі частоти в короткому вікні часу, або низькочастотний компонент у широкому вікні часу, але не обидва коливання одночасно. Вейвлет-функції для різних діапазонів частот використовують часові вікна відповідної тривалості, які обернено пропорційні значенню частот, і в результаті розтягування або стиснення, а так само і зсуву за часом, дозволяють проводити аналіз обох компонентів.

Базова форма згортки, що характеризує взаємозв'язок двох функцій у частотній області, визначається за формулою

$$y(n) = \sum_{k=0}^{N-1} x(k) \cdot g(n-k), \quad (2.9)$$

де  $y(n)$  – значення вихідних коефіцієнтів апроксимації вейвлет-функції;

$x(k)$  – початкові значення елементів зображення;

$g(n)$  – коефіцієнти вейвлет-фільтра (коефіцієнти вейвлет-функції).

Одержання набору коефіцієнтів масштабуючої функції вейвлет-перетворення здійснюється за формулою

$$y_L(n) = \sum_{k=0}^{N-1} x(k) \cdot h(2n-k), \quad (2.10)$$

де  $y_L(n)$  – значення низькочастотної складової вейвлет-трансформанти або значення вихідних укрупнених масштабуючих коефіцієнтів апроксимації вейвлет-функції;

$h(n)$  – коефіцієнти низькочастотного вейвлет-фільтра або коефіцієнти укрупненої масштабуючої вейвлет-функції.

Одержання набору уточнюючих деталізуючих коефіцієнтів вейвлет-перетворення здійснюється за формулою

$$y_H(n) = \sum_{k=0}^{N-1} x(k) \cdot g(2n - k), \quad (2.11)$$

де  $y_H(n)$  – значення високочастотної складової вейвлет-трансформанти або значення вихідних деталізуючих коефіцієнтів апроксимації вейвлет-функції;

$g(n)$  – коефіцієнти високочастотного вейвлет-фільтра або коефіцієнти деталізуючої вейвлет-функції.

Отже, проводячи вейвлет-фільтрацію по рядках і стовпцях, одержуємо квадратури вейвлет-перетворення (рис. 2.6).

$LL$	$HL$
$LH$	$HH$

Рис. 2.6. Одержання квадратур вейвлет-перетворення при обробці зображень по рядках і стовпцях

Виходить, послідовно апроксимуючи вейвлет-функціями значення елементів, розташованих по рядках та стовпцях, одержуємо квадратури вейвлет-перетворення, які містять значення укрупнених і деталізуючих масштабуючих коефіцієнтів.

Формування коефіцієнтів  $LL$  квадратури вейвлет-трансформанти орієнтується на значення, що отримані в результаті обчислень (2.10), які проводилися по рядках, і в результаті застосування аналогічної процедури для стовпців

$$y_{LL}(n) = \sum_{k=0}^{M-1} y_L(n) \cdot h(2n - k). \quad (2.12)$$

Коефіцієнти  $LH$  квадратури вейвлет-трансформанти так само утворюються при обробці рядків (3.14), але для стовпців застосовується вже виділення уточнюючих деталізуючих коефіцієнтів вейвлет-перетворення

$$y_{LH}(n) = \sum_{k=0}^{M-1} y_L(n) \cdot g(2n - k). \quad (2.13)$$

Формування коефіцієнтів  $HL$  квадратури вейвлет-трансформанти орієнтується на значення, що отримані в результаті обчислень (2.11) по рядках, а для стовпців застосовується виділення масштабуючих коефіцієнтів вейвлет-перетворення

$$y_{HL}(n) = \sum_{k=0}^{M-1} y_H(n) \cdot h(2n - k). \quad (2.14)$$

Для одержання коефіцієнтів *HH* квадратури вейвлет трансформанти проводиться розрахунок значень деталізуючих коефіцієнтів по рядках (2.11). У подальшому для стовпців так само застосовується виділення уточнюючих деталізуючих коефіцієнтів вейвлет-перетворення

$$y_{HH}(n) = \sum_{k=0}^{M-1} y_H(n) \cdot g(2n - k). \quad (2.15)$$

Послідовність фільтрації кожної квадратури по рядках і стовпцях початкового зображення обумовлює їх особливості, які визначають специфічний склад значень елементів квадратур, що несуть інформацію тільки про окремі частотні складові усього частотного діапазону, який займають значення початкового зображення (рис. 2.7).

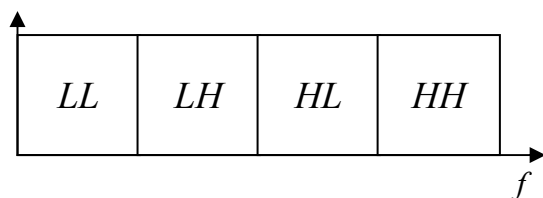


Рис. 2.7. Подання квадратур вейвлет-перетворення в частотній області

Коефіцієнт кореляції сусідніх елементів зображення, що описує статистичні зв'язки між яскравостями цих елементів, близький до 1. Отже, знаючи яскравість одного елемента, можна з високим ступенем імовірності передбачити ступінь яскравості сусіднього, наприклад, вважаючи їх просто рівними. Вейвлет-перетворення здатне перетворити масив відліків зо-

браження в матрицю коефіцієнтів, які будуть мати менший рівень кореляції, та дозволяє домогтися локалізації даних за рахунок розкладання на корельовані і некорельовані коефіцієнти. При цьому для слабонасичених зображень значна частина коефіцієнтів вейвлет-перетворення буде мати малий динамічний діапазон.

Вейвлет-перетворення може застосовуватися як до всього зображення, так і до його фрагментів. Якщо потрібно забезпечити якість не всього зображення, а лише його частини, даний фрагмент піддається вейвлет-перетворенню; інші фрагменти зображення лише грубо апроксимуються малим набором апроксимуючих коефіцієнтів. Це дозволяє забезпечити економію в обсязі переданих даних і скоротити час виконання операцій обчислень і передачі.

Практично квантування виконується шляхом поелементного ділення матриці коефіцієнтів ДВП на матрицю квантування, значення елементів якої зростають у міру віддалення від лівого верхнього кута та наближення до правого нижнього кута. При декодуванні на приймальній стороні коефіцієнти матриці ДВП помножуються на елементи матриці квантування, відновлює вірні значення коефіцієнтів, але з помилкою округлення, величини якої незначні для низькочастотних компонентів блока відліків зображення, але великі для високочастотних. Або з погляду часової області – значення помилки округлення незначне для укрупнених масштабуючих коефіцієнтів передбачення вейвлет-перетворення, але велике для деталізуючих. Цей аспект так само пов'язаний з поділом значень коефіцієнтів по ділянках динамічного діапазону:

$$\varsigma_{i,j} = y_{i,j} - [y_{i,j}],$$

де  $[ ]$  – операція виділення цілої частини.

Перекручування квантування необоротні, однак, якщо компресія не надмірно велика, для ока вони можуть бути малопомітними.

Внаслідок невеликої значущості *HH*-квадратури трансформанти ДВП (значення елементів трансформанти лежать у малому діапазоні величин і більшість із них для класу слабонасичених зображень дорівнює нулю) у деяких механізмах стиснення її не враховують (стандарти JPG, JPG-2000), скорочуючи при цьому обсяг даних і відповідно збільшуючи ступінь стиснення. Однак це негативно позначається на деталізуючих ознаках зображення, тому що для насичених деталями зображень високочастотні деталізуючі складові виділяють контури та границі реконструйованого зображення.

У ході розгляду елементів квадратур трансформанти ДВП можна зменшити динамічний діапазон за рахунок негативних елементів. Для цього необхідно скласти матрицю бітових значень, у яку заносяться стани елементів квадратур трансформанти (за винятком *LL*-квадратури, оскільки вона не містить негативних елементів). Надалі по цих бітових значеннях можна точно визначити, який елемент мав позитивне значення, а який негативне.

### ***Контрольні питання та завдання***

1. Назвіть методи трансформування відеозображень в технологіях ефективного кодування.
2. Які перетворення є ортонормованими?
3. Які базиси є синусоїдальними?

4. Назвіть особливості трансформування на основі дискретно-косинусного перетворення.

5. Які базиси називають кусочно-постійними?

6. Назвіть особливості дискретного перетворення Хаара.

7. Назвіть особливості дискретного перетворення Уолша.

8. Назвіть особливості вейвлет-перетворення зображень.

9. Дайте визначення такого поняття, як трансформація.

10. Дайте визначення такого поняття, як трансформоване зображення.

11. Дайте визначення такого поняття, як базисна функція.

12. Назвіть основні принципи побудови ортогональних перетворень.

13. Що розуміється під чутливістю компонент трансформант?

14. Що розуміється під глобальною чутливістю компонент трансформант?

15. Що розуміється під локальною чутливістю компонент трансформант?

16. Назвіть основні показники ортогональних перетворень.

17. Які відмінності має перетворення Фур'є у порівнянні з дискретним косинусним перетворенням?

18. Які переваги має дискретне косинусне перетворення у порівнянні з іншими ортогональними перетвореннями?

19. Назвіть особливості трансформанти дискретного косинусного перетворення?

20. Які недоліки має дискретно-косинусне перетворення?

21. Що розуміється під несинусоїдальними ортогональними перетвореннями?

## РОЗДІЛ 3

# БАЗОВІ ТЕХНОЛОГІЇ ТА МЕТОДИ КОДУВАННЯ ВІДЕОІНФОРМАЦІЇ

### 3.1. Універсальні коди. Методи кодування за алфавітом джерела інформації

Припускають наявність простих типів закономірностей, обумовлених обмеженнями на потужність алфавіту джерела повідомлень або алфавіту в рамках конкретного повідомлення.

Такі прості обмеження можуть відноситися до статистичного, структурного або комбінаторного типів закономірностей, що виявляються.

Код Бодо і монотонні поелементні коди відносяться до класу кодування, що враховує усунення закономірностей відносно динамічного діапазону в конкретному фрагменті. Причому такі закономірності можуть бути обумовлені або структурними обмеженнями в конкретному або в трансформованому фрагменті зображення. Для початкового фрагмента обмеження на динамічний діапазон залежать від значень яскравості у фрагменті зображення. Чим менше яскравість, тим менше динамічний діапазон.

Для трансформованих зображень – чим більше насиченість, тим більше низькочастотні і високочастотні складові та середня яскравість

#### Код Бодо

Простий поелементний код Бодо передбачає наявність відомостей про:

– розміри елементів машинної пам'яті (у разі подання відеоданих для зберігання на борту);

– максимальне значення  $a_{\max}$  елементів фрагмента зображення, яке обробляється.

Якщо заданий фрагмент  $A$  елементів  $a_{ij}$  зображення, тобто  $A = \{a_{ij}\}$ ,  $i = \overline{1, m}$ ,  $j = \overline{1, n}$ , то простий код Бодо формується у два етапи:

*Етап 1.* Відшукується максимальне значення елементів для фрагмента  $A$ , для чого використовується формула

$$a_{\max} = \max_{i,j}(a_{ij}).$$

*Етап 2.* Визначається кількість  $l_{\max}$  біт, яка потрібна для подання максимального елемента  $a_{\max}$ , що задається співвідношенням

$$l_{\max} = [\log_2 a_{\max}] + 1.$$

*Етап 3.* Величина  $l_{\max}$  записується на початку кодового подання і є службовою інформацією, яка вказує на границі кодового опису сусідніх елементів зображення. Після чого під кожен елемент зображення відводиться код довжиною, яка дорівнює  $l_{\max}$  біт.

Простий блоковий код Бодо полягає в поданні в кожному кодовому слові декількох елементів початкового фрагмента зображення. Наприклад, така ситуація виникає при поданні

декількох елементів кодованої послідовності в одному машинному слові (один регістр зовнішньої пам'яті).

Узагальнений код Бодо передбачає наявність додаткових відомостей, а саме:

- мінімального значення в рядку (стовпці)  $\mu_i = \min_{1 \leq j \leq n} \{a_{ij}\}$ ;
- максимального значення в стовпці (рядку)  $\lambda_j = \max_{1 \leq i \leq m} \{a_{ij}\}$ ;
- кроку дискретизації  $\delta_i$  елементів  $i$ -го рядка.

Довжина узагальненого коду Бодо для елементів  $a_{ij}$  масиву  $A$  визначається за формулою

$$l_i = \lceil \log_2(\lambda_i - \mu_i) / \delta_i \rceil + 1 \text{ [біт]}.$$

Зрозуміло, що для відновлення всіх елементів масиву  $A$  потрібно додатково передавати (зберігати) три вектори, а саме вектор мінімальних та максимальних значень відповідно в рядках і стовпцях кодованого масиву, а також вектор дискретизації елементів у рядках.

### Монотонні поелементні коди

Якщо об'єм алфавіту дуже великий (наприклад, як алфавіт розглядаються не окремі букви, а пари, трійки букв), це означає що зберігання і модифікація інформації про кожну букву потребує значних витрат пам'яті і часу. У цих випадках достатньо ефективними виявляються дуже прості методи універсального кодування [3; 18 – 20].

Послідовність букв джерела перетворюватиметься в послідовність натуральних чисел, тому нам знадобляться префіксні коди нескінченного об'єму.

Префіксний код безлічі натуральних чисел  $N = \{1, 2, \dots\}$  ми називатимемо *монотонним кодом*, якщо для будь-яких  $i, j \in N$ ,  $i < j$  довжини відповідних кодових слів  $l_i$  та  $l_j$  задовольняють нерівності  $l_i \leq l_j$ .

Приведемо декілька прикладів монотонних кодів.

### Унарний код

Нагадаємо, що запис вигляду  $0^m$  або  $1^m$  означає відповідно серію з  $m$  нулів або одиниць. Унарний код зіставляє числу  $i$  двійкову комбінацію вигляду  $1^{i-1}0$  або  $0^{i-1}1$ . Наприклад, унарними кодами чисел 1, 2 і 3 є послідовності  $\text{unar}(1) = 0$  або  $\text{unar}(1) = 1$ ,  $\text{unar}(2) = 10$  або  $\text{unar}(2) = 01$ , та  $\text{unar}(3) = 110$  або  $\text{unar}(3) = 001$  відповідно. Довжина кодового слова для числа  $i$  дорівнює  $l_i = i$  бітам. Приклад унарного кодування подано в табл. 3.1 [18].

Таблиця 3.1

### Приклади унарних кодів

$n$	1	2	3	4	5	$n$
$\text{unar}(n)$	1	01	001	0001	00001	$\underbrace{00\dots0}_n n$ $n-1$

Неважко перевірити, що унарний код оптимальний, якщо числа  $i$  розподілені за геометричним законом, а саме

$$p_i = (1 - \alpha)^{i-1} \alpha, \quad i = 1, 2, \dots$$

з параметром  $\alpha = 1/2$ .

Унарний код (табл. 3.4) як різновид монотонних кодів застосовується для кодування послідовностей, елементи яких мають великий об'єм алфавіту та для кодування недвійкових даних двійковим арифметичним кодером, тобто для переведення недвійкових послідовностей у двійкові з подальшим їх арифметичним кодуванням.

Для значень  $\alpha < 1/2$  більш ефективнішими є код Голомба.

### Код Голомба – Райса

Введемо параметр коду  $m$ , де  $T = 2^m$ . Код Голомба [21] для числа  $i$  складається з двох частин. Перша частина – це унарний код числа  $\lfloor i/T \rfloor$ , друга частина – двійковий запис у вигляді послідовності довжини  $m$  остачі від ділення  $i$  на  $T$ , а саме

$$i \rightarrow \underbrace{000\dots 1}_{\text{unary}(\lfloor \frac{i}{T} \rfloor + 1)} i \bmod(T).$$

Тут  $i \bmod(T)$  – остача від ділення  $i$  на  $T$ ;  $\lfloor \frac{i}{T} \rfloor$  – ціла ча-

стина результату ділення  $\frac{i}{T}$ .

Очевидно, що довжина коду Голомба для числа  $i$  дорівнює  $l_i = \lfloor i/T \rfloor + m$ . Наприклад, при  $m = 3$  кодом Голомба числа 21 буде послідовність 001101, де

$$\left\lfloor \frac{21}{2^3} \right\rfloor + 1 = 3, 3_{(2)} \rightarrow 001;$$

$$21 \bmod(2^3) = 5, 5_{(2)} \rightarrow 101.$$

Тут  $l_{(2)}$  – двійковий запис десяткового числа  $l$ .

### Код Галлагера – ВанВухріса

Узагальненням коду Голомба – Райса на випадок, коли саме задається параметр  $T$ , та він не є ступенем двійки, постає код Галлагера – ВанВухріса [18; 21; 22]. Якщо  $\log T$  – не ціле число, то для подання деяких (менших) значень остачі від ділення  $i$  на  $T$  використовується  $m = \lfloor \log T \rfloor$  біт, для інших (великих) – послідовність  $(m + 1)$  біт. Наприклад, при  $T = 5$  кодovими словами для значень остач 0, 1, 2, 3 та 4 будуть послідовності  $\{00\}$ ,  $\{01\}$ ,  $\{10\}$ ,  $\{110\}$  та  $\{111\}$  відповідно.

Код Галлагера – ВанВухріса оптимальний, якщо параметр  $T$  пов'язаний з параметром геометричного розподілу  $\alpha$  співвідношенням

$$(1 - \alpha)^T + (1 - \alpha)^{T+1} \leq 1 < (1 - \alpha)^{T-1} + (1 - \alpha)^T. \quad (3.1)$$

Завдяки надзвичайно простій схемі кодування і досить високій ефективності коди Голомба – Райса та Галлагера – ВанВухріса часто застосовують при кодуванні аудіо- та відеосигналів. Для застосування в схемах універсального кодування вони не забезпечують потрібного ефекту: по-перше, потрібно знати (або оцінювати) значення параметра  $m$  або  $T$ , по-друге, довжина кодового слова  $l_i$ , як і для унарного коду, лінійно зростає зі збільшенням числа  $i$  та для деяких розподілів кодування може бути неефективною.

### Код Левенштейна

Код Левенштейна [23] простіше всього пояснити на конкретному прикладі. Припустимо, що потрібно передати число  $i = 21$ . Двійкове подання цього числа має вигляд  $\{10101\}$ . Безпосередньо використовувати при кодуванні двійкові подання натуральних чисел не можна, тому що такий код не буде префіксним. Найпростіший вихід полягає в тому, щоб приписати на початку слова префікс, який вказує довжину двійкового запису числа (у даному випадку це число 5). Якщо це число закодувати префіксним, наприклад унарним, кодом і приписати зліва до двійкового запису числа, то код вийде однозначно декодованим. У даному прикладі для числа 21 одержимо кодове слово  $\{11110\ 10101\}$  або  $\{00001\ 10101\}$ , тобто

$$i \rightarrow \underbrace{000\dots1}_{\text{unar}(\lceil \log_2 i \rceil + 1)} \lceil \log_2 i \rceil + 1 \text{ або } i \rightarrow \underbrace{111\dots0}_{\text{unar}(\lceil \log_2 i \rceil + 1)} \lceil \log_2 i \rceil + 1.$$

У загальному випадку довжина двійкового подання буде дорівнювати  $2(\lceil \log_2 i \rceil + 1) = 2\lceil \log_2 i \rceil$ .

Крок за кроком покращуватимемо цей спосіб кодування. Відмітимо, що першою значущою цифрою двійкового запису числа буде завжди 1 або 0. У цьому випадку, наприклад, одиницю можна не передавати, декодер сам допише відсутню одиницю, якщо знатиме довжину двійкового запису числа. Позначимо через  $\text{bin}'(i)$  двійковий запис натурального числа  $i$  без першої одиниці; прямими дужками позначається довжина двійкової послідовності. Отже, простий монотонний код числа  $i$  має таку структуру:

$$\text{mon}(i) = (\text{unar}(\lceil \text{bin}'(i) \rceil + 1), \text{bin}'(i)). \quad (3.2)$$

Довжини кодів слів дорівнюють

$$l_i = 2\lceil \log_2 i \rceil + 1. \quad (3.3)$$

Щоб зробити запис ще коротше, слід додатково обробити довжину двійкового запису отриманого числа та передати його значущі розряди (окрім першої одиниці). Потім додатково обробляється довжина двійкового запису числа значущих розрядів і т. д. Ітерації продовжуються, доки не залишиться один значущий розряд. Щоб декодування було однозначним, достатньо приписати префікс, що містить закодовану префіксним кодом кількість ітерацій. Зазначимо, що мінімальна кількість ітерацій дорівнює 0 (при кодуванні "1"). Тому як префіксний код можна вибрати унарний код збільшеної на 1 кількості операцій. Одержане кодове слово буде кодовим словом *коду Левенштейна*.

Наприклад, для числа 21 обчислюється  $\text{bin}'(21) = 0101$ , потім  $\text{bin}'(4) = 00$ ,  $\text{bin}'(2) = 0$ . Кількість ітерацій дорівнює 3, тому кодове слово коду Левенштейна має вигляд

$$\text{lev}(21) = (1110)(0)(00)(0101) = 11100000101.$$

Декодер коду Левенштейна, декодуючи унарний код, отримує інформацію, що ітерацій було 3. Прочитавши один значущий розряд  $\{0\}$  і дописавши до нього в початок 1, декодер одержує послідовність  $\{10\}$ . Це означає, що на передостанній ітерації довжина числа була 2. Прочитавши 2 розряди і

дописавши зліва 1, декодер одержує послідовність {100}. Тепер декодер прочитує 4 розряди і дописує зліва 1. Виходить послідовність {10101}, їй відповідає число 21. Оскільки це вже остання 3-тя ітерація, то число 21 є результатом декодування.

У табл. 3.2 наведені кодові слова, одержані за правилом (3.2), і кодові слова коду Левенштейна для деяких чисел натурального ряду і довжини кодових слів.

Таблиця 3.2

**Монотонні коди деяких чисел натурального ряду**

$i$	Монотонний код		Код Левенштейна		Код Елайєса	
	$\text{mon}(i)$	$l_i$	$\text{lev}(i)$	$l_i$	$\text{elias}(i)$	$l_i$
1	0	1	0	1	0	1
2	100	3	100	3	100	3
3	101	3	101	3	101	3
4	11000	5	110000	6	110000	6
...	...	5	...	6	...	6
7	11011	5	110011	6	110011	6
8	1110000	7	110100	7	110100	7
...	...	7	...	7	...	7
15	1110111	7	111011171	7	111011171	7
16	111100000	9	111000009111000	11	1110000000	10
...	...	9	...	11	...	10
31	111101111	9	11100001111	11	1110001111	10
32	11111000000	11	111000100000	12	11100100000	11
...	...	...	...	12	...	11
63	11111011111	11	111000111111	12	11100111111	11
$2^{15}$	$1^{15}00^{15}$	33	$1^4 0^{11} 1^{11} 10^{15}$	26	$1^5 0^{11} 1^{11} 0^{15}$	25
$2^{102}$	$1^{1023} 00^{1023}$	2047	$1^4 0^{1001} 1^9 0^{1023}$	1041	$1^{10} 0^9 1^9 0^{1023}$	1043

**Спрощений код Левенштейна (код Елайєса)**

Опишемо простіший код, наведений у роботі Елайєса [22]. Числу  $i=1$  припишемо кодове слово  $\text{elias}(1) = 0$ . Для чисел  $i > 1$  кодові слова обчислюються за таким правилом:

$$\text{elias}(i) = (\text{unar}(|\text{bin}'(|\text{bin}'(i))|) + 2, \text{bin}'(|\text{bin}'(i)|), \text{bin}'(i)). \quad (3.4)$$

Кодове слово складається з трьох частин. Справа (у третій частині) записуємо двійкове подання числа без першої одиниці. Їй передує друга частина, в якій записано двійкове подання довжини третьої частини теж без першої одиниці. У першій частині записаний унарний код збільшеної на дві довжини другої частини кодового слова.

Наприклад, для числа 21 кодове слово має вигляд

$$\text{elias}(21) = (1110)(00)(0101) = 1110000101.$$

Довжина кодового слова коду Елайєса для довільного числа  $i$  дорівнює

$$l_i = \begin{cases} 1, & \rightarrow i = 0; \\ \lfloor \log i \rfloor + 2 \lfloor \log \lfloor \log i \rfloor \rfloor + 2, & \rightarrow i > 1, \end{cases} \quad (3.5)$$

отже, задовольняє нерівності

$$l_i \leq \log i + 2 \log(1 + \log i) + 2, \quad i = 1, 2, \dots \quad (3.6)$$

У другому доданку під знаком логарифма додана одиниця, щоб надати сенсу цьому виразу при  $i = 1$ .

Щоб чисельно порівняти три монотонні коди, звернемося до табл. 3.2. Простий код (3.2) у більшості випадків дає найкращий результат, але для цього коду довжина кодових слів зростає зі збільшенням змінної  $i$  швидше, ніж для двох інших кодів. Це видно, зокрема, з порівняння формул (3.3) і (3.4). Коди Левенштейна і Елайеса практично еквівалентні; виграш коду Левенштейна виявляється тільки при астрономічно великих значеннях  $i$ .

### 3.2. Методи кодування відеоданих з урахуванням виявлення ймовірно-статистичних закономірностей

Даний матеріал присвячено розгляду методів стискаючого кодування на основі виявлення ймовірно-статистичних закономірностей. Розглядаються базові моделі статистичних джерел інформації. Для таких типів джерел вводяться способи оцінки кількості інформації. Викладаються базові методи кодування послідовностей, що породжуються статистичними джерелами інформації. Скорочення надмірності досягається в результаті виявлення обмежень відповідно до заданої моделі. Якщо обмеження, що виявляються, відносяться до ймовірно-статистичних, то відповідно й надмірність є ймовірно-статистичною [11; 24; 25; 26].

### Статистичні моделі джерел дискретних повідомлень

*Джерела з рівноймовірними і статистично незалежними символами в повідомленнях.* При розгляді простого джерела було встановлено, що його повідомлення математично моделюються дискретною випадковою послідовністю  $\{a_{ij}\}$ . Використання математичної моделі для вивчення реального явища припускає обов'язковим переклад на мову математики властивостей досліджуваного явища; в даному випадку необхідно формалізувати правила складання дискретним джерелом повідомлень довжини  $n$  на своєму виході.

Що стосується простого джерела, то правила складання його повідомлень гранично прості: будь-який символ алфавіту в повідомленні може бути використаний незалежно від інших елементів і абсолютно рівноправний з усіма іншими символами алфавіту.

Статистичного аналізу символів у таких послідовностях проводити не потрібно, оскільки у рівноймовірних статистично незалежних символів умовна ймовірність дорівнює безумовним і визначається як  $p(a_i / a_j) = p(a_i) = \frac{1}{M}$ , тобто залежить лише від величини потужності алфавіту  $M$ . З цієї точки зору вважатимемо, що повідомлення такого джерела піддаються нуль-вимірному статистичному аналізу; самі джерела називатимемо джерелами нульового типу або джерелами Хартлі, а величини їх інформаційних заходів відзначатимемо підрядковим індексом "0".

Для прикладу проведемо математичний опис декількох простих дискретних джерел:

– двійкового джерела:

$$M = 2; a_1 \hat{=} "1"; a_2 \hat{=} "0"; n = 5; p(a_1) = p(a_2) = \frac{1}{2};$$

– трійкового джерела:

$$M = 3; a_1 \hat{=} "1"; a_2 \hat{=} "0"; a_3 \hat{=} "-1"; n = 5; p(a_1) = p(a_2) = p(a_3) \frac{1}{3}.$$

Таким чином, для математичного опису простих джерел достатньо вказати кількість символів у їх алфавіті  $M$  і довжину повідомлення  $n$ .

*Джерела з нерівноймовірними і статистично незалежними символами в повідомленнях.* Існують інші правила формування повідомлень з елементів алфавіту.

Припустимо, наприклад, що алфавіт джерела має потужність  $M$ , але його здатність генерувати різні символи алфавіту різна, а саме елемент  $a_1$  може генерувати необмежено; елементів  $a_2$  джерело здатне створити за час своєї роботи дещо менше, ніж елементів  $a_1$ , елементів  $a_3$  – ще менше і т. д. Очевидно, що у джерела повідомлень дуже великої довжини ймовірність появи різних букв різна, навіть якщо вибір елементів алфавіту для формування повідомлення проводився статистично незалежно.

Математичною моделлю таких повідомлень буде дискретна випадкова послідовність, елементи якої нерівноймовірні, але вибір даного елемента проводиться в незалежних від попередніх виборів умовах. Такі послідовності в математиці відомі як послідовності Бернуллі.

Їх повний математичний опис проводиться на основі одновимірного статистичного аналізу безумовної ймовірності появи символів алфавіту в повідомленнях  $p(a_i)$ . Аналогічно дискретні джерела, символи алфавіту яких у повідомленнях з'являються статистично незалежно, але з різною ймовірністю, називатимемо одновимірними джерелами або джерелами Бернуллі, а величини їх інформаційних заходів відзначатимемо підрядковим індексом "1" (наприклад  $I_1, H_1$ ).

Для математичного опису джерел Бернуллі необхідно задати їх алфавіт у вигляді ймовірнісного ансамблю

$$\begin{aligned} A(\dot{a}_1, \dot{a}_2, \dot{a}_3, \dots, \dot{a}_M) &\hat{=} \{a_i\}, \quad i = \overline{1, M}, \\ D\{\dot{\delta}(\dot{a}_1), \dot{\delta}(\dot{a}_2), \dots, \dot{\delta}(\dot{a}_M)\} &\hat{=} \{\delta(\dot{a}_i)\}, \quad i = \overline{1, M} \end{aligned}$$

і вказати довжину повідомлень, що формуються на їх виходах. Зрозуміло, що кількість  $V_1$  різних повідомлень, яка може бути створене джерелом Бернуллі на виході при відомих його параметрах  $M$  та  $n$ , вже не буде дорівнювати  $M^n$ , як у джерел Хартлі, а буде дещо меншою. Отже, виникає необхідність встановлення несуперечливих способів визначення кількості інформації до повідомлень джерел Бернуллі.

*Джерела з нерівноймовірними і статистично залежними символами в повідомленнях.* До цього типу дискретних джерел відноситимемо такі джерела, повідомлення на виході яких формуються при нерівноймовірній і залежній вибірці елементів алфавіту джерела. Дискретні випадкові послідовності з такими властивостями вивчав російський математик А. А. Марков, ім'ям якого і названі джерела з даними правилами формування повідомлень.

Причини, що приводять до появи нерівномірності елементів алфавіту в повідомленні, були розглянуті при описі джерел Бернуллі. Зупинимося коротко на причинах появи взаємної залежності елементів у повідомленні.

У російській мові, письмовий текст якої є характерним прикладом джерела Маркова, залежність елементів у словах обумовлена обмеженнями, що накладаються російською граматиною на правила формування повідомлень. Так, у граматиці російської мови виключено написання букви “Ъ” після букви “Ъ” і не допускається написання, наприклад, після букви “Ч” букв “Ы”, “Я”, “Ю” і т. п.

Обмеження такого вигляду можуть інтерпретуватися як двовимірна зв’язність в послідовності Маркова (або ланцюги, як іноді називають послідовностями). Обмеження можуть розповсюджуватися на три і більше символів повідомлень дискретного джерела. У цих випадках математичними моделями повідомлень будуть двозв’язні, тривз’язні і більшої зв’язності ланцюги Маркова. Природно, що при цьому ускладнюються методи визначення інформаційних характеристик повідомлень порівняно з джерелами першого і другого типів.

Для математичного опису джерел Маркова необхідно, окрім значення параметрів  $M$  та  $n$ , повністю задати ймовірнісний ансамбль, як і при описі джерел Бернуллі, і матрицю переходів кожного попереднього стану джерела в такий:

$$\begin{pmatrix} p(a_1/a_1) & p(a_2/a_1) & \dots & p(a_M/a_1) \\ p(a_1/a_2) & p(a_2/a_2) & \dots & p(a_M/a_2) \\ \dots & \dots & \dots & \dots \\ p(a_1/a_M) & p(a_2/a_M) & \dots & p(a_M/a_M) \end{pmatrix} \begin{matrix} \{a_i\}, i = \overline{1, M}; \\ \{p(a_i)\}, i = \overline{1, M}; \\ \dots \\ \cong |p(a_j/a_i)| \begin{matrix} i = \overline{1, M} \\ j = \overline{1, M} \end{matrix} \end{matrix}$$

Якщо ймовірність переходу  $p(a_j/a_i)$  залежить лише від того, яка подія спостерігалася в один з попередніх моментів часу, то послідовність символів і джерело повідомлень, яке вона описує, називатимемо двозв’язними ланцюгами Маркова, а якщо вона залежить від декількох попередніх подій, то послідовність і джерело називатимемо багатозв’язними ланцюгами Маркова. Якщо умови формування повідомлень дискретним джерелом не змінюються в часі, то дискретні джерела називатимемо стаціонарними дискретними джерелами.

Дискретні джерела Маркова є найбільш загальними моделями джерел дискретних повідомлень, одновимірні (Бернуллі) і нуль-вимірні (Хартлі) моделі є їх окремими випадками.

При вивченні інформаційної якості джерел необхідно вміти визначати кількість інформацій, що містяться в повідомленнях джерела; середню кількість інформації, яка переноситься одним довільним елементом алфавіту даного джерела; кількість інформації, що створюється на виході джерелом в одиницю часу; вміти інформаційно порівнювати повідомлення різних джерел, визначаючи надмірність одного по відношенню до іншого і т. д.

Для порівняння між собою різних джерел повідомлень необхідно ввести кількісну міру, яка дала б можливість об’єктивно оцінити інформацію, що міститься в повідомленні. Така міра вперше була введена К. Шенноном, а потім формалізована О. Я. Хінчиним.

### **Визначення кількісної міри джерел дискретних повідомлень**

У 1865 р. німецький фізик Рудольф Клаузіус увів у статистичну фізику поняття ентропії або міри врівноваженості сис-

теми. У 1921 р. засновник більшої частини математичної статистики англієць Роналд Фішер уперше ввів термін “інформація” в математику. У 1948 р. Клод Шеннон у своїх роботах із теорії зв’язку виписує формули для обчислення кількості інформації і ентропії. Термін “ентропія” використовується Шенноном за порадою патріарха комп’ютерної ери фон Неймана, який зазначив, що отримані Шенноном формули для розрахунку кількості інформації в теорії зв’язку збіглися з відповідними формулами статистичної фізики, а також те, що “точно ніхто не знає”, що ж таке ентропія.

Розглянемо основи інформаційного підходу Шеннона.

Будь-яка інформація доходить до споживача після прийому повідомлення, тобто в результаті дослідження. Повідомлення, що отримується на приймальній стороні, несе корисну інформацію лише в тому випадку, якщо є невизначеність щодо стану джерела. Якщо дослід може закінчитися тільки одним результатом і спостерігач заздалегідь знає результат дослідження, то за його результатом він не отримує ніякої інформації. Наприклад, якщо повідомлять, що по каналу зв’язку передається зображення, яке є квадратом, забарвленим чорним кольором, то ніякої інформації це зображення не надасть, оскільки наперед відомо зміст цього зображення. У такій події, як передача по каналу зв’язку заздалегідь відомого зображення, немає нічого невизначеного, ймовірність цієї події дорівнює одиниці і кількість інформації, що переноситься повідомленням про таку подію, дорівнює нулю. Інформація з’явиться лише тоді, коли джерело матиме принаймні більше одного можливого стану.

Розглянемо джерело, що видає послідовність незалежних дискретних повідомлень  $\{a_i\}$ , кожне з яких випадковим чином вибирають з алфавіту повідомлення  $A = \{a_1, a_2, \dots, a_M\}$ , де  $M$  –

розмір алфавіту джерела. Таке джерело називатимемо джерелом без пам’яті з кінцевим дискретним алфавітом.

У кожному елементарному повідомленні  $a_i$  для його одержувача міститься деяка інформація. Визначимо кількісну міру цієї інформації і з’ясуємо, від чого вона залежить.

В одержувача інформації завжди існує більша або менша невизначеність щодо того, яке повідомлення  $a_i$  з можливих буде передано.

Абсолютно очевидно, що ступінь цієї невизначеності або несподіванки передачі  $a_i$ , залежить від імовірності передачі того або іншого повідомлення. Наприклад, якщо ймовірність передачі будь-якого повідомлення  $a_i$  дуже висока, то ще до передачі ми майже напевно знаємо, яке повідомлення буде передано, і його прийом не надасть нам майже ніякої нової інформації.

Таким чином, очевидно, що кількість інформації, що міститься в елементарному повідомленні  $a_i$ , є деякою функцією від імовірності передачі цього повідомлення  $p(a_i)$ :

$$F(a_i) = f\{p(a_i)\}.$$

Визначимо вигляд цієї функції  $f$  за умови, що міра кількості інформації  $F(a_i)$  задовольняє двом інтуїтивним властивостям.

1. Якщо вибір повідомлення  $a_i$  заздалегідь зумовлений ( $p(a_i) = 1$  – невизначеності немає), то кількість інформації в цьому повідомленні дорівнює нулю, тобто  $F(a_i) = f\{1\} = 0$ .

2. Якщо джерело послідовно вибирає повідомлення  $a_i$  та  $a_j$  і ймовірність такого вибору  $p(a_i; a_j)$  є сумою ймовірностей подій  $a_i$  та  $a_j$ , то кількість інформації в цих двох елементарних повідомленнях дорівнюватиме сумі кількостей інформації в кожному з них.

Ймовірність  $p(a_i; a_j)$  сумісної появи подій  $a_i$  та  $a_j$ , як відомо, визначається за формулою повної ймовірності

$$p(a_i; a_j) = p(a_i)p(a_j / a_i).$$

Тоді має виконуватися умова

$$f\{p(a_i)p(a_j / a_i)\} = f\{p(a_i)\} + f\{p(a_j / a_i)\}.$$

Неважко здогадатися, що функцією, яка задовольняє цим двом умовам, є функція вигляду

$$F(a_i) = \alpha \log p(a_i).$$

При цьому як коефіцієнт  $\alpha$ , так і основа логарифма можуть бути вибрані довільно. Проте для зручності (щоб кількісна міра інформації була позитивною) приймають  $\alpha = -1$ . Основу логарифма зазвичай вибирають рівною двом, і тоді

$$F(a_i) = -\log_2 p(a_i).$$

Визначена таким чином одиниця вимірювання інформації називається *двійковою одиницею*, або *бітом інформації*.

Кількість інформації, що міститься в одному елементарному повідомленні  $a_i$ , ще ніяк не характеризує джерело. Одні елементарні повідомлення можуть нести багато інформації, але передаватися дуже рідко, інші – передаватися частіше, але нести менше інформації. Тому джерело може бути охарактеризовано середньою кількістю інформації, що припадає на одне елементарне повідомлення. Таке джерело має назву *ентропія джерела* і визначається таким чином:

$$H_1 = -\sum_{i=1}^M p(a_i) \cdot \log p(a_i), \quad i = \overline{1, M}.$$

Ентропія як кількісна міра інформативності джерела має такі властивості.

1. Ентропія є величиною дійсною, обмеженою і ненегативною.

2. Статистична ентропія для детермінованих повідомлень дорівнює нулю, тобто  $H_1 = 0$ , якщо хоча б одне з повідомлень має ймовірність, яка дорівнює одиниці.

3. Ентропія максимальна, якщо повідомлення  $a_i$  рівноймовірні, тобто

$$p(a_1) = p(a_2) = \dots = p(a_M),$$

і тоді  $H_1 = \log M$ .

Як видно з останнього виразу, у разі рівноймовірних повідомлень ентропія зростає зі збільшенням об'єму алфавіту джерела (зростанням кількості повідомлень). При нерівно-

ймовірних елементарних повідомленнях  $a_i$  ентропія відповідно зменшується.

4. Ентропія двійкового джерела ( $M = 2$ ) може змінюватися від нуля до одиниці.

5. Якщо для двох ансамблів  $A$  та  $Y$  розподілами ймовірності є однакові набори чисел (відрізняються тільки порядком проходження елементів), то  $H(A) = H(Y)$ .

6. Якщо ансамблі  $A$  і  $Y$  незалежні, то сумісна ентропія дорівнює  $H(A Y) = H(A) + H(Y)$ .

7. Ентропія – це опукла  $\cap$  функція розподілу ймовірності на елементах ансамблю  $A$ .

8. “Вирівнювання” ймовірності елементів ансамблю приводить до збільшення ентропії.

9. Взаємодозначна обробка інформації не приводить до збільшення ентропії.

### **Інформаційні характеристики джерел залежних повідомлень**

Для джерел Хартлі і Бернуллі вважається, що символи в повідомленнях на виходах цих джерел некорельовані, тобто поява наступних символів повідомлення статистично не пов’язана з попередніми символами. Для складання математичних моделей таких джерел виявилось достатнім мати в своєму розпорядженні лише безумовну ймовірність вибору символів алфавіту джерела як елемент повідомлень.

У реальних джерелах повідомлень незалежність елементів повідомлення один від одного є надзвичайно рідкісним явищем.

При побудові математичної моделі, наприклад для російського тексту, необхідно, окрім безумовної ймовірності вибо-

ру букв а, б, в, ..., я як елементи повідомлень, ураховувати умовну ймовірність вибору після букви а будь-якої букви алфавіту:  $p(a/a)$ ,  $p(б/a)$ , ...,  $p(я/a)$ . Так само потрібно діяти при визначенні умовної ймовірності вибору букв алфавіту після букв б, в. Для побудови адекватнішої російському тексту моделі необхідно враховувати зв’язки, що існують між трійками, четвірками, п’ятірками букв у текстах і т. д. Зрозуміло, що у міру наближення моделі до дійсного опису всіх особливостей російської граматики кількість імовірнісних параметрів для моделі буде надзвичайно збільшуватися. У зв’язку з цим, до сьогодні не проводилися розрахунки інформаційних характеристик текстових повідомлень з урахуванням зв’язків, що розповсюджуються більш ніж на п’ять-шість елементів.

Припустимо, що, крім нерівноймовірності появи в повідомленні символів алфавіту, достатньо врахувати зв’язки, які існують між сусідніми елементами повідомлень, тобто, що модель джерела може бути повністю описана за допомогою ймовірностей  $\{p(a_i)\}$  безумовного вибору символів алфавіту для складання повідомлень й умовних ймовірностей  $\{p(a_j/a_i)\}$  вибору символів  $a_i$  після символів  $a_j$ . Отже, як і раніше, можна прийняти, що кількість елементів у повідомленні дорівнює  $n$ , а потужність алфавіту джерела –  $M$ .

Спочатку при вказаних обмеженнях зі всіх можливих повідомлень аналізованого двозв’язного джерела Маркова розглянемо тільки такі, які починаються з деякого конкретного символу  $a_i$ . Оскільки при заданому першому символі повідомлення наступні символи матимуть ймовірність  $p(a_j/a_i)$ , можна обчислити умовну ентропію джерела (якщо він створює тільки повідомлення, які починаються з букви  $a_i$ ) за фор-

мулою джерела Бернуллі, тому що накладенням обмеження на початок повідомлення виключається залежність усіх елементів повідомлення, які надходять через один від першого. Величина умовної ентропії буде дорівнювати

$$H_i = -\sum_{j=1}^M p(a_j / a_i) \log p(a_j / a_i).$$

Оскільки поява елемента  $a_i$  в повідомленнях є все ж таки випадковою подією, то необхідно зняти обмеження, накладене припущенням про цілком конкретний початок повідомлення, і лише після цього можна буде визначити повну ентропію даного джерела Маркова. Для зняття обмеження усереднимо величину умовної ентропії по всіх можливих початкових буквах, імовірності вибору яких дорівнюють  $p(a_i)$ . Усереднена величина  $M(H_i)$  і дорівнюватиме повній інформаційній продуктивності двозв'язного марковського дискретного джерела:

$$H_2 = M(H_i) = -\sum_{i=1}^m p(a_i) \sum_{j=1}^m p(a_j / a_i) \log p(a_j / a_i).$$

Оскільки величина

$$p(a_i)p(a_j / a_i) = p(a_i a_j)$$

є ймовірністю поєднання символів  $a_i$  та  $a_j$ , вираз для інформаційної продуктивності марковського джерела можна записати в такому вигляді:

$$H_2 = -\sum_{i=1}^M \sum_{j=1}^M p(a_i a_j) \log p(a_j / a_i).$$

Якщо всі елементи двозв'язного марковського джерела рівномірні, тобто  $p(a_i) = p(a_j) = \frac{1}{M}$ , то

$$H_2 = -\frac{1}{M} \sum_{i=1}^M \sum_{j=1}^M p(a_j / a_i) \log p(a_j / a_i).$$

Неважко переконатися, що за наявності взаємозв'язків між елементами повідомлення ентропія джерела буде менша, ніж за відсутності їх. Дійсно, поклавши в попередньому виразі  $p(a_j / a_i) = p(a_j)$  (що відповідає відсутності взаємозв'язку між  $a_i$  та  $a_j$ ), одержимо величину

$$H_i = -\sum_{j=1}^M p(a_j) \log p(a_j),$$

яка чисельно дорівнює інформаційній продуктивності джерела Бернуллі.

Можна перетворити формули для випадків, коли в повідомленнях є зв'язки не тільки між парами символів, але й між  $s$  символами ( $s > 2$ ) у групі. Для опису джерел з такими зв'язками буде потрібна модель не простої, а складної марковської послідовності.

За наявності зв'язків між трійками символів повідомлень повинні розглядатися поєднання  $(a_i; a_j; a_k)$ , а інформаційна продуктивність джерела Маркова може бути обчислена за формулою

$$H_3 = - \sum_{i=1}^M \sum_{j=1}^M \sum_{k=1}^M p(a_i; a_j; a_k) \log p(a_k / a_i; a_j).$$

У загальному випадку за наявності зв'язків між  $s$  символами повідомлення обчислення інформаційної продуктивності необхідно здійснювати за формулою

$$H_s = - \underbrace{\sum_{i=1}^M \sum_{k=1}^M \dots \sum_{l=1}^M}_{s \text{ разів}} p(a_i; a_j; \dots; a_l) \log p(a_i / a_l; \dots; a_k).$$

Знаючи величину інформаційної продуктивності джерела і враховуючи, що вона показує кількість інформації, яка переноситься в середньому одним елементом повідомлення даного джерела, неважко обчислити кількість інформації, що міститься в  $n$ -значному повідомленні даного дискретного джерела.

Кількість інформації для двозв'язного марковського джерела буде дорівнювати

$$I_2 = nH_2 = -n \sum_{i=1}^M \sum_{j=1}^M p(a_i; a_j) \log(a_j / a_i),$$

а для  $s$ -зв'язного –

$$I_s = nH_s.$$

Важливою властивістю ентропії марковських джерел є те, що її величина зменшується у міру збільшення ступеня зв'язності символів у повідомленнях джерела. Ураховуючи цю властивість, а також властивість ентропії джерела повідомлень з незалежними елементами, можна записати нерівності

$$H_0 \geq H_1 \geq H_2 \geq \dots \geq H_s.$$

Наприклад, розглянемо джерело двійкових повідомлень, елементи 0 та 1 якого в повідомленні з'являються незалежно один від одного з імовірністю  $p(0) = p(1) = 0,5$ . Інформаційна продуктивність такого джерела дорівнює

$$H_0 = \log_2 2 = - \sum_{i=1}^2 0,5 \log_2 0,5 = 1 \text{ [біт/елемент]}.$$

Наприклад, якщо у двійкового джерела  $p(0) = 0,3$ ,  $p(1) = 0,7$ , то його інформаційна продуктивність дорівнює

$$H_1 = -0,3 \log_2 0,3 - 0,7 \log_2 0,7 = 0,88 \text{ [біт/елемент]}.$$

*Приклад.* Джерелу властиві такі ймовірнісні зв'язки:  $p(0) = 0,125$ ;  $p(1) = 0,875$ ;  $p(0/0) = 0,3$ ;  $p(1/0) = 0,7$ ;  $p(1/1) = 0,1$ ;  $p(0/1) = 0,9$ .

Тоді

$$\begin{aligned}
H_2 &= -\sum_{i=1}^2 p(a_i) \sum_{j=1}^2 p(a_j / a_i) \log p(a_j / a_i) = \\
&= -0,125(0,3 \log_2 0,3 + 0,7 \log_2 0,7) - \\
&- 0,875(0,1 \log_2 0,1 + 0,9 \log_2 0,9) = 0,51 \text{ [дв.од./симв.]}.
\end{aligned}$$

Якби зв'язки були відсутні, тобто виконувалася б рівність

$$\begin{aligned}
p(0/0) &= p(0/1) = p(0) = 0,125; \\
p(1/1) &= p(1/0) = p(1) = 0,875,
\end{aligned}$$

то

$$\begin{aligned}
H_2 &= -0,125(0,125 \log_2 0,125 + 0,875 \log_2 0,875) - \\
&- 0,875(0,125 \log_2 0,125 + 0,875 \log_2 0,875) = \\
&= -(0,125 \log_2 0,125 + 0,875 \log_2 0,875) \cdot (0,125 + 0,875) = \\
&= H_1 = 0,543 \text{ [дв.од./симв.]}.
\end{aligned}$$

Наведені приклади чисельно підтверджують зменшення інформаційної продуктивності (ентропії) джерел дискретних повідомлень при появі нерівномірності та у міру збільшення ступеня зв'язності символів у повідомленнях джерел.

Отже, за наявності зв'язку між елементарними повідомленнями ентропія джерела знижується, причому тим більше, чим сильніше зв'язок між елементами повідомлення.

Таким чином, можна зробити певні висновки щодо ступеня інформативності джерел повідомлень.

1. *Ентропія джерела і кількість інформації тим більше, чим більше розмір алфавіту джерела.*

2. Ентропія джерела залежить від статистичних властивостей повідомлень і має максимальне значення, якщо повідомлення джерела рівноймовірні і статистично незалежні.

3. Ентропія джерела, що виробляє нерівноймовірні повідомлення, завжди менше максимально досяжної.

4. За наявності статистичних зв'язків між елементарними повідомленнями (пам'яті джерела) його ентропія зменшується.

### Нерівномірне кодування дискретних джерел

Джерела інформації надмірні в тому сенсі, що при ефективному кодуванні можна зменшити витрати на передачу або зберігання породжуваної ними інформації. Для подання даних буде потрібно тим менше біт, чим менше ентропія. Це означає, що для ефективного кодування зручними є джерела, в яких деякі букви (послідовності букв) мають значно більшу ймовірність, ніж інші букви (послідовності). Для таких джерел потрібно використовувати кодування, яке формує короткі кодові комбінації для повідомлень, що часто зустрічаються, і, навпаки, довгі кодові комбінації – для повідомлень, що зустрічаються рідше.

Для кодування окремих букв джерела оптимальним побуквеним (поелементним) кодом є відомий код Хаффмана.

#### **Постановка завдання нерівномірного побуквеного кодування**

Припустимо, що для деякого дискретного джерела  $A$  з відомим розподілом імовірності  $\{p(a), a \in A\}$  потрібно побудувати ефективний нерівномірний двійковий код над алфавітом  $B = \{b\}$ , а також, що  $B = \{0, 1\}$ . По-перше, всі ідеї повною мірою ілюструються на цьому прикладі, по-друге, узагальнення

у разі довільного алфавіту не є складним. Окрім цього на практиці для кодування джерел використовуються майже виключно двійкові коди.

Як приклад розглянемо азбуку Морзе, в якій кожній букві відповідає комбінація крапок “•” і тире “–”. Наприклад, букві “e”, що часто зустрічається, відповідає комбінація “•”, а рідкіснішій букві “q” відповідає комбінація “– – – •”. Проте, простою заміною крапки нулями, а тире – одиницями, ефективного коду не отримаємо, оскільки не вистачить пауз, що розділяють символи всередині букв (ця пауза відповідає інтервалу часу, який дорівнює часу передачі крапки), пауз, що розділяють букви (3 крапки), пауз, що розділяють слова (7 крапок). Отже, код Морзе – це недвійковий код.

Нам необхідний такий двійковий код, який допускає однозначний поділ послідовності кодограм на окремі кодові слова без використання будь-яких додаткових символів. Цю вимогу називають властивістю однозначного декодування. Нерівномірний побуквений код  $C = \{c\}$  об’ємом  $|C|$  над алфавітом  $B$  визначається як довільна множина послідовностей однакової або різної довжини з букв алфавіту  $B$ . Код є однозначно декодованим, якщо будь-яка послідовність символів  $B$  єдиним способом розбивається на окремі кодові слова.

*Приклад 3.1.* Для джерела  $A = \{0; 1; 2; 3\}$  серед чотирьох кодів

$$C_1 = \{0\,0, 0\,1, 1\,0, 1\,1\}; \quad C_2 = \{1, 0\,1, 0\,0\,1, 0\,0\,0\}; \\ C_3 = \{1, 1\,0, 1\,0\,0, 0\}; \quad C_4 = \{0, 1, 1\,0, 0\,1\}$$

перші три коди однозначно декодуються, останній код не декодується.

Першим кодом цього прикладу є рівномірний код, який може бути однозначно декодований.

Для декодування другого коду можна застосувати стратегію, за якою декодер прочитує символ за символом і кожного разу перевіряє, чи не збігається отримана послідовність з одним з кодових слів. У разі успіху відповідне повідомлення видається одержувачу, і декодер починає декодування наступного повідомлення. При застосуванні коду  $C_2$  неоднозначностей бути не може, оскільки жодне слово не є продовженням іншого.

Якщо жодне кодове слово не є початком іншого, то код називається префіксним. Префіксні коди є однозначно декодованими.

Префіксною множиною двійкових послідовностей  $W$  називається кінцева множина двійкових послідовностей, таких, що жодна послідовність цієї множини не є префіксом або початком ніякої іншої послідовності множини  $W$ . Код  $C_3$  свідомо не префіксний, проте він є однозначно декодованим. Кожне слово коду  $C_3$  отримано переписуванням у зворотному порядку відповідного слова коду  $C_2$ . Для декодування послідовності кодових слів коду  $C_3$  можна переписати прийняту послідовність у зворотному порядку і для декодування використовувати декодер коду  $C_2$ . Таким чином, можна зробити висновок, що префіксність є достатньою, але не необхідною умовою однозначного декодування.

Графічно зручно подавати префіксні коди у вигляді кодових дерев. Зокрема, кодове дерево коду  $C_2$  для прикладу 3.1 зображено на рис. 3.1.

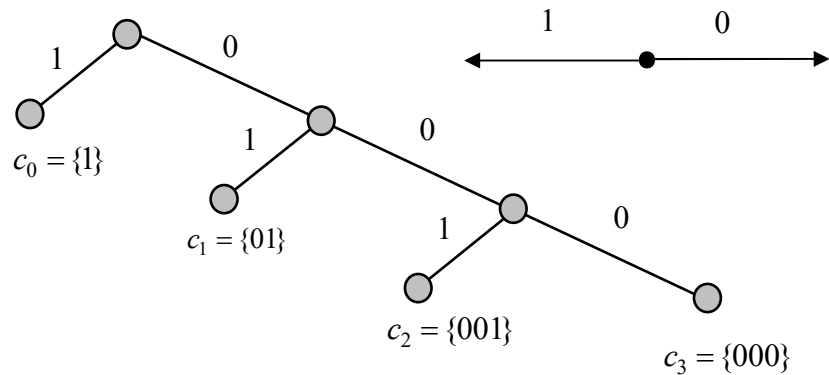


Рис. 3.1. Кодове дерево коду  $C_2$  для прикладу 3.1

Вузли дерева розміщуються ярусами. На початковому (нульовому) ярусі розташовано один вузол, називаний коренем дерева. Вузли наступних ярусів пов'язані з вузлами попередніх ярусів ребрами. Для двійкового коду з кожного вузла виходить не більше двох ребер. Ребрам приписані кодові символи. У цьому прикладі прийнято, що правому ребру приписується символ 0, а лівому – символ 1. Таким чином, кожній вершині дерева відповідає послідовність, яка сканується вздовж шляху, що пов'язує даний вузол з коренем дерева.

Вузол називається кінцевим, якщо з нього не виходить жодного ребра. Код називають деревоподібним, якщо як кодові слова він містить тільки кодові слова, які відповідають кінцевим вершинам кодового дерева.

Опис коду як дерева та префіксність є синонімами в тому сенсі, що всякий деревоподібний код є префіксним, і всякий префіксний код може бути поданий за допомогою кодового дерева. Необхідно з'ясувати, як впливає умова префіксності на побудову оптимальних кодів. Адже в результаті накладення

умови префіксності звужується клас кодів, серед яких ми шукаємо найкращий. Далі переконаємося в тому, що відповідь на це питання негативна.

Обговоримо тепер, які саме префіксні коди вважати найкращими для кодування. Оскільки сама мета розгляду нерівномірного кодування полягала в зменшенні витрат на передачу повідомлень, то логічно вибрати як критерій якості коду середню довжину кодових слів. Розглянемо джерело  $A = \{1, \dots, M\}$ , яке породжує букви з імовірністю  $\{p_1, \dots, p_M\}$ . Припустимо, що для кодування букв джерела вибрано код  $C = \{c_1, \dots, c_M\}$  з довжинами кодових слів, які відповідно дорівнюють  $\text{length}(c_1) = l_1, \dots, \text{length}(c_M) = l_M$ . Середньою довжиною кодових слів називається величина

$$\bar{l} = M(l_i) = \sum_{i=1}^M p_i l_i.$$

Ще один важливий аспект, який має бути враховано при порівнянні способів нерівномірного кодування, – це складність реалізації кодування і декодування. Залежно від сфери застосування може змінюватись і значення допустимої складності, та саме поняття складності. Наприклад, при реалізації алгоритму на універсальному комп'ютері практично відсутнє обмеження на розмір оперативної пам'яті, але має значення обчислювальна складність. При реалізації кодера та декодера у вигляді інтегральної мікросхеми або за допомогою спеціалізованого сигнального процесора визначальним параметром, навпаки, є пам'ять.

Поелементні коди часто використовуються як складова комплексних алгоритмів. У цих випадках має значення не лише складність власне кодування і декодування елемента, а й складність побудови або модифікації коду при зміні статистичних даних про джерело.

Отже, завдання поелементного нерівномірного кодування формулюється як завдання побудови однозначно декодованого коду з найменшою середньою довжиною кодів слів при заданих обмеженнях на складність.

### Оптимальний поелементний код – код Хаффмана

Запропонований у 1952 р. студентом Девідом Хаффманом алгоритм побудови оптимальних нерівномірних кодів – одне з базових досягнень теорії інформації як з теоретичної, так і з прикладної точок зору.

Розглянемо ансамбль повідомлень  $A = \{1, \dots, M\}$  з імовірністю повідомлень  $\{p_1, \dots, p_M\}$ . Без втрати спільності вважається, що повідомлення впорядковані за спаданням імовірності, тобто  $p_1 < p_2 < \dots < p_M$ . Завдання полягає в побудові оптимального коду, тобто коду з найменшою можливою середньою довжиною кодів слів. Зрозуміло, що при заданій імовірності такий код може бути не єдиним; можливе також існування сімейства оптимальних кодів. Установимо деякі властивості всіх кодів цього сімейства. Ці властивості підкажуть нам шлях до знаходження одного з оптимальних кодів.

Нехай двійковий код  $C = \{c_1, \dots, c_M\}$  із довжинами кодів слів  $\{l_1, \dots, l_M\}$  оптимальний для даного ансамблю повідомлень.

*Властивість 3.1.* Якщо  $p_i < p_j$ , то  $l_i \geq l_j$ .

*Властивість 3.2.* Не менше двох кодів слів мають однакову максимальну довжину  $l_M = \max_{1 \leq i \leq M} \{l_i\}$ .

*Властивість 3.3.* Серед кодів слів довжиною  $l_M = \max_{1 \leq i \leq M} \{l_i\}$  знайдуться два слова, які відрізняються лише одним останнім символом.

Перш ніж сформулювати наступну властивість, введемо додаткові позначення. Для даного ансамблю  $A = \{1, \dots, M\}$  і деякого коду  $C$ , що задовольняє властивостям 3.1 – 3.3, введемо допоміжний ансамбль  $A' = \{1, \dots, M-1\}$ , повідомленням якого зіставимо ймовірність  $\{p'_1, \dots, p'_{M-1}\}$  таким чином:

$$p'_1 = p_1; \quad p'_{M-2} = p_{M-2}; \quad p'_{M-1} = p_{M-1} + p_M.$$

З коду  $C$  побудуємо код  $C'$  для ансамблю  $A$ , приписавши повідомленням  $a'_1, \dots, a'_{M-2}$  ті ж кодові слова, що і для коду  $C$ , тобто  $c'_i = c_i$ ,  $i = \overline{1, M-2}$ , а повідомленню  $x'_{M-1}$  – слово  $c'_{M-1}$ , що є загальною частиною слів  $c_{M-1}$  та  $c_M$  (згідно з властивістю 3.3 ці два кодові слова відрізняються тільки одним останнім символом).

*Властивість 3.4.* Якщо код  $C'$  для повідомлення  $A'$  оптимальний, то і код  $C$  оптимальний для повідомлення  $A$ .

Отже, сформульовані властивості оптимальних префіксних кодів зводять завдання побудови коду об'єму  $M$  до завдання побудови коду об'єму  $M' = M-1$ . Це означає, що ми отримали рекурентне правило побудови кодового дерева оптимального нерівномірного коду.

## Побудова статистичного коду методом Хаффмана

Вихідні повідомлення	Імовірності вихідного та перетворених наборів							Код повідомлення
	I	II	III	IV	V	VI	VII	
					0,42	0,58 0,42	1	
$a_7$	0,34	0,34	0,34	0,34 0,24	0,34 0,24			11
$a_6$ $a_5$	0,23 0,19	0,23 0,19	0,23 0,19 0,14	0,23 0,19				01 00
$a_4$ $a_3$	0,1 0,07	0,1 0,07	0,1					100 1010
$a_2$ $a_1$	0,06 0,01							10110 10111

Алгоритм побудови двійкового статистичного коду Хаффмана передбачає виконання таких операцій. Повідомлення, які належать кодуванню, упорядковують відповідно до спадання чи зростання їх імовірності. Два початкові повідомлення, що мають найменшу ймовірність, умовно замінюють одним, приписуючи йому сумарну ймовірність об'єднаних повідомлень. Таким чином, одержують допоміжний набір із  $(M - 1)$  повідомлень, який аналогічним чином перетвориться в допоміжний набір з  $(M - 2)$  повідомлень. Перетворення необхідно продовжувати до тих пір, доки весь початковий набір із  $M$  належних кодуванню повідомлень не перетвориться на одне допоміжне повідомлення з імовірністю, яка дорівнює 1.

Приклад подібних перетворень для семи початкових повідомлень, що мають розподіл імовірності

$$p(a_1) = 0,01; \quad p(a_2) = 0,06; \quad p(a_3) = 0,07; \quad p(a_4) = 0,1;$$

$$p(a_5) = 0,19; \quad p(a_6) = 0,23; \quad p(a_7) = 0,34,$$

зведений у табл. 3.3 та наведений на рис. 3.2.

*Недоліки методу Хаффмана*

Найбільшою складністю кодів Хаффмана є необхідність мати таблиці ймовірності для кожного типу кодованих даних. Це не вважається проблемою, якщо відомо, наприклад, що стисненню підлягає англійський або російський текст.

У цьому випадку для кодера і декодера підбирається відповідне для англійського або російського тексту кодове дерево. Узагалі ж, якщо ймовірність символів для вхідних даних невідома, статичні коди Хаффмана працюють неефективно.

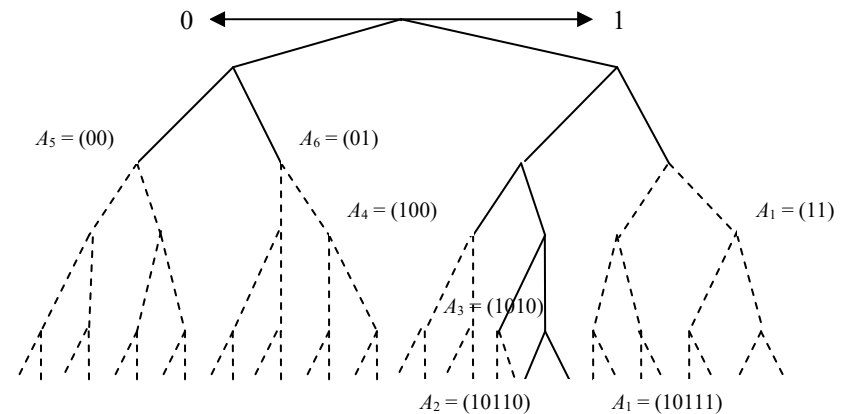


Рис. 3.2. Кодове дерево для коду Хаффмана

Вирішенням цієї проблеми є статистичний аналіз кодованих даних, який виконується в ході першого проходу за даними, і побудова на його основі кодового дерева. Кодування при цьому виконується другим проходом.

Існує також динамічна версія стиснення Хаффмана, яка може будувати дерево Хаффмана “на льоту” під час зчитування і активної компресії. Дерево постійно оновлюється, щоб відображати зміни ймовірності вхідних даних. Проте й вона на практиці має серйозні обмеження і недоліки і, крім того, забезпечує меншу ефективність стиснення.

Ще одним недоліком коду Хаффмана є те, що мінімальна довжина кодового слова не може бути меншою одного біта на елемент. Навпаки, ентропія повідомлення цілком може становити і 0,1, і 0,01 біт/елемент. У цьому випадку код Хаффмана стає істотно надмірним. Проблема вирішується застосуванням алгоритму до блоків символів, але тоді ускладнюється процедура кодування/декодування і значно розширюється кодове дерево, яке потрібно зберігати разом з кодом.

Код Хаффмана забезпечує середню довжину коду, яка збігається з ентропією, тільки у тому випадку, якщо ймовірність символів джерела є цілими негативними ступенями двійки, наприклад  $1/2 = 0,5$ ;  $1/4 = 0,25$ ;  $1/8 = 0,125$ ;  $1/16 = 0,625$  і т. д. На практиці ж така ситуація зустрічається дуже рідко або може бути створена блокуванням символів.

Розглянута реалізація коду Хаффмана відноситься до нерівномірних поелементних (побуквених) кодів без пам'яті. У цьому випадку код формується для окремих елементів без урахування інформації про інші елементи. Передбачається, що елементи в послідовностях з'являються з різною ймовірністю незалежно один від одного.

Якщо розглядається стаціонарне джерело і його розподіл імовірності на елементах не змінюється від елемента до елемента, то будь-який з описаних вище способів кодування може бути використаний для кодування окремих повідомлень джерела. У багатьох випадках саме такий підхід застосовується на практиці як найпростіший. У той же час, можна виділити клас застосувань, для яких поелементне кодування буде свідомо неоптимальним.

Відповідно до основної теореми статистичного кодування без наявності перешкод напрямком для вирішення даної проблеми є врахування пам'яті джерела інформації в процесі кодування. Це потенційно може підвищити ефективність кодування. Для такого варіанта можливі два підходи, а саме:

1) будувати блоковий код, вибравши достатньо велику довжину блоків  $n$ . Блоковий код можна формувати, наприклад, з використанням поелементного кодування, застосовуючи його для цілих послідовностей;

2) використовувати статистичну модель джерела інформації, в якій враховуватимуться залежності між елементами послідовності. Наприклад, для цього можна використовувати методи апроксимації джерела ланцюгом Маркова достатньо високого порядку.

В основі таких підходів лежить припущення, яке полягає в тому, що для початкової послідовності  $\{a_1, a_2, \dots, a_n\}$ ,  $a_i \in A = \{a\}$ , що генерується дискретним стаціонарним джерелом, існує ймовірнісний опис, тобто можна обчислити всі багатовимірні розподіли ймовірності та відповідно ентропію на повідомлення  $H = H(X)_\infty$ .

*Розглянемо перший підхід.* Блоковий код ділить початкову послідовність на блоки заданої довжини, а потім кожен блок

Таблиця кодової відповідності для блокового коду

Таблиця кодера	
Початковий блок	Кодове слово
01	0
10	10
11	11

Замінюючи кожен блок присвоєним йому кодовим словом з таблиці кодера, отримаємо таку кодову комбінацію

$$C = \{0; 1\ 0; 1\ 0; 0; 1\ 1; 1\ 0; 0; 0\}.$$

Довжина кодової комбінації дорівнює 12 бітам. Це означає, що середня кількість двійкових розрядів, яка приходить на один елемент початкової послідовності, буде дорівнювати 0,75 бітам. Це менше ніж ентропія джерела для випадку, коли враховується тільки нерівномірність розподілу ймовірності появи елементів. Такий вигравш досягається, оскільки для блоків враховуються закономірності не окремих елементів, а цілих послідовностей (блоків).

У той же час слід зазначити, що при оцінці середніх витрат блокового коду на один елемент початкової послідовності не враховувались витрати на кодову таблицю, об'єм якої збільшуватиметься пропорційно розміру блоків. Для наведеного прикладу кількість елементів укрупненого алфавіту ("гіпералфавіту") в три рази більше, ніж початкової двійкової послідовності.

замінюють кодовим словом з множини кодованого алфавіту. Отриману послідовність кодових слів об'єднують у результуючий кодовий рядок на виході кодера. Якщо всі блоки мають довжину, яка дорівнює  $n$ , то блоковий код вважають кодом  $n$ -го порядку.

Розглянемо приклад формування блокового коду для двійкової послідовності

$$A = \{0\ 1\ 1\ 0\ 1\ 0\ 0\ 1\ 1\ 1\ 1\ 0\ 0\ 1\ 0\ 1\}.$$

Кількість інформації, що міститься в одному елементі такої послідовності, з урахуванням нерівномірності ймовірності їх появи дорівнює 0,9887 біта.

Припустимо, що блоковий код має порядок, який дорівнює двом. Відповідно до цього розіб'ємо послідовність на блоки по два елементи в кожному і отримаємо  $\{0\ 1\}$ ;  $\{1\ 0\}$ ;  $\{1\ 0\}$ ;  $\{0\ 1\}$ ;  $\{1\ 1\}$ ;  $\{1\ 0\}$ ;  $\{0\ 1\}$ ;  $\{0\ 1\}$ .

Розглянемо ці блоки як елементи нового укрупненого алфавіту ("гіпералфавіту"), тобто  $\{(0\ 1); (1\ 0); (1\ 1)\}$ . Щоб визначити, який код призначити якому з символів цього нового алфавіту, знайдемо вектор частот появ елементів додаткового алфавіту в послідовності блоків. Отримаємо вектор частот  $\{p_{01} = 4; p_{10} = 3; p_{11} = 1\}$ . Блок  $\{0\ 1\}$  з'являється чотири рази, наступний блок  $\{1\ 0\}$  зустрічається три рази, блок  $\{1\ 1\}$  з'являється лише один раз.

Використовуючи для отримання компактного подання, наприклад, код Хаффмана, за вищерозглянутим алгоритмом отримаємо кодову відповідність (табл. 3.4).

## Приклад коду з кінцевою пам'яттю

Елемент, попередній елемент	Кодове слово
(128, -)	1
(130, 128)	0
(132, 128)	10
(124, 128)	11
(128, 125)	1
(125, 130)	1
(128, 132)	1
(128, 124)	1

Результатом кодування буде вектор  $C(A) = \{1011101111011\}$  завдовжки в 13 біт та зі швидкістю стиснення  $R = 13/11 = 1,18$  біт/елемент послідовності, тоді як у разі кодування рівномірним трьохрозрядним кодом з  $R = 3$  біт/елемент на кодове подання початкової послідовності знадобиться 33 біти.

Основна перешкода на шляху застосування цього кодування полягає в експоненціальному зростанні складності при збільшенні довжини блоків  $n$ . Пояснимо цю проблему таким простим прикладом.

Припустимо, що кодуванню підлягають півтонові зображення. Елементи зображення мають об'єм алфавіту  $|A| = 2^8 = 256$ . При кодуванні послідовностей довжини  $n = 2$  об'єм алфавіту зростає до  $|A^2| = 2^{16} = 65536$ . Далі при  $n = 3$  і  $n = 4$  об'єми алфавітів будуть відповідно дорівнювати  $2^{24} = 16777216$  і  $2^{32} = 4294967296$ . Зрозуміло, що працювати з кодами таких розмірів неможливо.

Розглянемо тепер другий підхід, в основі якого лежить припущення про наявність статистичної залежності між елементами початкових послідовностей. Кількість елементів, для яких існує така залежність, визначає порядок моделі. Наприклад, якщо існує статистична залежність між  $n$  елементами послідовності, то модель джерела таких послідовностей описується марковським ланцюгом  $n$ -го порядку. Порядок моделі передбачається заданим.

Код з кінцевою пам'яттю при кодуванні початкової послідовності  $\{a_1, a_2, \dots, a_n\}$  використовує кодову книгу, що складається з декількох різних кодів без пам'яті. Кожна вибірка даних кодується кодом без пам'яті з кодової книги шляхом визначення деякої кількості попередніх вибірок даних.

Приклад 3.2. Припустимо, що кодована відеопослідовність має вигляд

$$A = \{128; 130; 125; 128; 132; 128; 124; 128; 130; 125; 128\}.$$

У цій послідовності абсолютно очевидна сильна статистична залежність між її суміжними елементами, яка має обов'язково враховуватися при виборі методу кодування.

Кодер з пам'яттю при кодуванні поточного елемента враховує значення попереднього елемента. Таким чином, кодове слово для поточного елемента 128 буде різним в поєднаннях (125; 128), (132; 128) і (124; 128), тобто код має пам'ять в один символ джерела (табл. 3.5).

Один з напрямків ефективного блокового кодування реалізується в технології арифметичного кодування, яке дозволяє ефективно кодувати блоки довжини  $n$  з надмірністю порядку  $2/n$  та зі складністю, зростаючою лише пропорційно квадрату довжини блока  $n$ . За рахунок малого програшу, яким нехтують, в швидкості коду складність може бути навіть лінійною по довжині коду. Ця перевага дозволяє все ширше використовувати арифметичне кодування в різноманітних системах обробки інформації.

### *Арифметичне кодування*

При арифметичному кодуванні [18; 19; 27; 28 – 43], на відміну від розглянутих нами методів, коли кодований символ (або група символів) замінюється відповідним йому кодом, результат кодування всього повідомлення представляється одним або парою дійсних чисел в інтервалі від 0 до 1. У процесі кодування початкового тексту інтервал, що відображає його, зменшується, а кількість десяткових (або двійкових) розрядів для його подання зростає. Чергові символи вхідного тексту скорочують величину інтервалу виходячи із значень їх імовірності, яка визначається моделлю. Імовірніші символи роблять це у меншій мірі, чим менш імовірні, і, отже, додають менше розрядів до результату.

Ідею алгоритму приписують Елайєсу, посилаючись на книгу Абрамсона, опубліковану в 1963 р. До кінця 70-х років минулого століття ця ідея не була затребувана. Декілька статей, опублікованих на рубежі 70-80-х років минулого століття мала радше теоретичний характер. Велику роль в розумінні як ідеї алгоритму, так і можливості його використання для вирішення практичних завдань відіграла стаття Уїттена, Ніла і Клі-

рі. Мабуть, саме цих учених слід вважати авторами алгоритму в його нинішній формі. У той же час, коли став відомим сам алгоритм, стало зрозуміло, що він є майже тривіальним узагальненням коду Шеннона для послідовності. Тому, наприклад, у книзі Ковера і Томаса арифметичне кодування називається алгоритмом Шеннона – Фано – Елайєса.

Розглянемо для простоти дискретне постійне джерело, що вибирає повідомлення з множини  $A = \{1, \dots, M\}$  з імовірністю  $\{p_1, \dots, p_M\}$ . Позначимо через  $\{q_1, \dots, q_M\}$  кумулятивну ймовірність повідомлень. Завдання полягає в кодуванні послідовностей множини  $A^{(n)} = \{A\}$ . При описі алгоритму кодування використовується позначення  $a(i; j)$  для короткого запису під-послідовності  $(a_i, \dots, a_j)$  у послідовності  $A = \{a_1, \dots, a_n\}$ .

Необхідно побудувати для ансамблю  $A^{(n)} = \{A\}$  достатньо простий та ефективний елементарний код. Спрощення полягає в тому, що ні кодер, ні декодер не зберігають та не формують усієї множини з  $|A^{(n)}|$  кодових слів. Замість цього при передачі конкретної послідовності  $A$  кодером обчислюється кодове слово  $C(A)$  тільки для даної послідовності  $A$ . Правило кодування зазвичай відоме декодеру і він відновлює повідомлення  $A$  по коду  $C(A)$ , не маючи повного списку кодових слів.

Можливими кандидатами на використання в такій схемі можна розглядати код Шеннона і код Гілберта – Мура. Проте використання коду Шеннона припускає впорядкування повідомлень за сподіванням імовірності. При великих  $n$  складність упорядкування виявиться неприпустимою для реалізації. Тому єдиним підходом залишається код Гілберта – Мура.

Відповідно до правила побудови коду Гілберта – Мура кодове слова формується по ймовірності  $p(a)$  та кумулятивній ймовірності  $q(a)$  як перші  $l(a) = -[\log_2 p(a)] + 1$  двійкові розряди після крапки в двійковому запису числа  $\sigma(a) = q(a) + p(a)/2$ .

Щоб обчислити  $q(a)$ , слід домовитися про деяку нумерацію послідовностей з  $A^{(n)}$ . Найбільш природний спосіб нумерації послідовностей полягає у використанні лексикографічної впорядкованості, яка на послідовностях позначатиметься знаком “ $\prec$ ”. Запис  $Y \prec A$  означатиме, що  $Y$  лексикографічно передує  $A$ .

Поняття лексикографічного порядку визначається таким чином. Для послідовностей довжини 1 (для окремих повідомлень з  $A$ ) вважатимемо, що повідомлення з меншим номером передує повідомленню з більшим номером. *Лексикографічний порядок* – це порядок, який зазвичай використовується при складанні словників.

Для двох послідовностей  $A = \{a_1, \dots, a_n\}$ ,  $Y = \{y_1, \dots, y_n\}$  позначимо через  $i$  найменший індекс, такий, за якого значення  $a_i \neq y_i$ . Тоді  $Y \prec A$ , якщо  $y_i \prec a_i$ , тобто  $y_i < a_i$ .

Отже, основне завдання полягає в обчисленні кумулятивної ймовірності

$$q(a) = \sum_{y \prec a} p(y),$$

оскільки для джерела без пам’яті ймовірності послідовностей  $p(a)$  обчислюються достатньо просто за формулою

$$p(a) = \prod_{i=1}^n p(a_i).$$

Виведемо рекурентну формулу для обчислення  $q(a)$ . Для цього виразимо ймовірність  $q(a_1^{(n)})$  через  $q(a_1^{(n-1)})$ . Наступний ланцюжок елементарних перетворень показує, як це можна зробити

$$\begin{aligned} q(a_1^{(n)}) &= \sum_{y_1^{(n)} \prec a_1^{(n)}} p(y_1^{(n)}) = \sum_{y_1^{(n-1)} \prec a_1^{(n-1)}} \sum_{y_n} p(y_1^{(n-1)} y_n) + \\ &+ \sum_{y_1^{(n-1)} = a_1^{(n-1)}} \sum_{y_n \prec a_n} p(y_1^{(n-1)} y_n) = \sum_{y_1^{(n-1)} \prec a_1^{(n-1)}} p(y_1^{(n-1)}) + \\ &+ \sum_{y_1^{(n-1)} = a_1^{(n-1)}} p(y_1^{(n-1)}) \sum_{y_n \prec a_n} p(y_n) = q(a_1^{(n-1)}) + p(a_1^{(n-1)}) q(a_n), \end{aligned}$$

де  $q(a_n)$  – кумулятивна ймовірність символу  $a_n$ .

Підсумуємо викладення у вигляді рекурентних формул

$$\begin{aligned} q(a_1^{(n)}) &= q(a_1^{(n-1)}) + p(a_1^{(n-1)}) q(a_n), \\ p(a_1^{(n)}) &= p(a_1^{(n-1)}) p(a_n). \end{aligned}$$

У цій рекурсії кожна пара значень  $(q(a_1^{(i)}), p(a_1^{(i)}))$  використовується рівно на одному кроці при обчисленні наступної пари  $(q(a_1^{(i+1)}), p(a_1^{(i+1)}))$ . Тому при реалізації арифметичного кодування новообчислені значення записуються в ті ж елементи пам’яті, в яких знаходилися попередні значення. У наведеному нижче алгоритмі кумулятивна ймовірність послідовнос-

## Кодування послідовності арифметичним кодом

Крок $i$	$a_i$	$p(a_i)$	$q(a_i)$	$F$	$G$
0	–	–	–	0,0000	1,0000
1	b	0,6	0,1	0,1000	0,6000
2	c	0,3	0,7	0,5200	0,1800
3	d	0,6	0,1	0,5380	0,1080
4	a	0,1	0,0	0,5380	0,0108
5	b	0,6	0,1	0,5391	0,0065
6	Довжина кодового слова		Кодове слово		
	$\lceil -\log G + 1 \rceil = 9$		$(F + G/2) \dots \rightarrow \hat{F} = 0,541 \rightarrow 100010101$		

У разі графічної інтерпретації процесу кодування на кожному кроці кодування перераховується початкова точка  $F$  і довжина  $G$  відрізка, в якому буде розташоване число, що відповідатиме кодів послідовності для заданої послідовності повідомлень. Так, після першого кроку ( $a_1 = b$ ) ми знаємо, що крапка буде розташована у відрізку  $[0,1; 0,7)$ . Оскільки друга буква  $a_2 = c$ , то початкова точка переміщується в значення 0,52, а довжина інтервалу зменшується до 0,18 і т. д. Після п'ятого кроку  $F = 0,5391$ . Додавши зсув  $G/2$  і округливши до 9 двійкових знаків, одержимо величину  $\hat{F} = 0,541$ . Тим самим вся послідовність повідомлень відобразиться в одній точці інтервалу  $[0; 1)$ . Як було показано вище, 9 розрядів достатньо для того, щоб послідовність була відновлена однозначно, тоб-

тей  $i = 1, 2, \dots$  зберігається у вигляді змінної  $F$ , ймовірність послідовностей  $i = 1, 2, \dots$  – у вигляді змінної  $G$ . Остаточне формулювання алгоритму може бути записане в такому вигляді.

**Алгоритм арифметичного кодування** передбачає виконання таких операцій.

1. Ініціалізація. За ймовірністю  $\{p_1, \dots, p_M\}$  повідомлень джерела обчислюється кумулятивна ймовірність

$$q_1 = 0, \text{ для } j = \overline{2, M}, \quad q_j = q_{j-1} + p_{j-1}.$$

Установлюються початкові значення допоміжних змінних  $F = 0, G = 1$ .

Від джерела приймається послідовність повідомлень  $a = (a_1, \dots, a_n)$ .

2. Для  $i = \overline{1, n}$  виконуються такі обчислення:

$$\begin{aligned} F &\leftarrow F + q(a_i)G, \\ G &\leftarrow p(a_i)G. \end{aligned}$$

3. Кодове слово для  $x$  формується як перші  $\lceil -\log G \rceil + 1$  розрядів після коми в двійковому записі числа  $(F + G/2)$ .

*Приклад 3.3.* Розглянемо джерело  $A = \{a; b; c\}$ , в якому ймовірність  $p_a = 0,1, p_b = 0,6, p_c = 0,3$ . Обчислення, виконані арифметичним кодером при кодуванні послідовності ( $A = bcbab$ ) довжини  $n = 5$ , наведені в табл. 3.6. У цій таблиці через  $\hat{F}$  позначено число  $(F + G/2)$ , округлене вниз з точністю до  $\lceil -\log G + 1 \rceil = 9$  двійкових розрядів.

то щоб найближча крапка, яка відповідає іншій послідовності повідомлень, була віддалена відносно  $\hat{F}$  на відстань, не менше  $1/2^9 = 1/512$ .

З опису алгоритму виходить, що на кожному кроці кодування виконується одне додавання та два множення. Звідси легко зробити висновок про те, що складність кодування послідовності з  $n$  повідомлень пропорційна  $n$ . Це невірно, оскільки на кожному кроці лінійно росте складність виконання самих операцій додавання і множення, що обумовлено наростанням кількості двійкових розрядів, необхідних для запису операндів.

Припустимо, що для подання ймовірності  $\{p_1, \dots, p_M\}$  використано числа розрядності  $d$ . Після першого кроку кодування точне подання  $F$  і  $G$  потребуватиме  $2d$  двійкові розряди. Відповідно після  $n$  кроків кодування кодер і декодер працюватимуть (у гіршому разі) з числами розрядності  $nd$  і, отже, сумарна складність має порядок

$$d + 2d + \dots + nd = \frac{n(n+1)d}{2}.$$

Таким чином, можна зробити висновок, що складність арифметичного кодування, пропорційна  $n^2$ .

Насправді, можлива практична реалізація арифметичного кодування зі складністю, пропорційною  $n$ .

Важлива особливість арифметичного кодування полягає в тому, що його можна адаптувати до випадку джерел з пам'яттю. Якщо, наприклад, як модель джерела розглядається простий ланцюг Маркова, то алгоритм кодування залишається

тим самим за винятком того, що замість одновимірної ймовірності  $p(a_i)$  та  $q(a_i)$  потрібно використовувати умовну ймовірність  $p(a_i | a_{i-1})$  та  $q(a_i | a_{i-1}) = \sum_{y \prec x_i} p(a_i | a_{i-1})$ .

### Деякі аспекти практичної реалізації арифметичного кодування

Як виходить з опису арифметичного кодування, при його практичній реалізації для кодування послідовностей великої довжини  $n$  виникають такі проблеми:

– арифметичне кодування вимагає значної точності обчислень, що призводить до неприпустимо високої складності реалізації;

– для формування кодового слова формально необхідна вся послідовність повідомлень, що призводить до неприпустимо великої затримки кодування, яка дорівнює довжині кодованої послідовності повідомлень.

Але обидві проблеми можна вирішити шляхом виключення з обчислень та поданням на вихід кодера тієї частини даних, яка не бере участі в подальших обчисленнях і, отже, не впливає на остаточний результат. Тим самим зменшується складність обчислень і затримка кодування. Пояснити це можна за допомогою аналізу роботи декодера коду Гілберта – Мура.

Нехай для джерела  $A = \{1, \dots, M\}$  відомі ймовірності  $\{p_1, \dots, p_M\}$ , за якими обчислені величини  $q_m = \sum_{j=1}^{m-1} q_j$ ,

$\sigma_m = q_m + p_m/2$ , довжини слів  $l_m = -\lceil \log_2 p_m / 2 \rceil + 1$  та кодові слова довжини  $l_m$ , отримані округленням величин  $\sigma_m$ . Округ-

лені до  $l_m$  розрядів після коми числа  $\sigma_m$  позначимо через  $\hat{\sigma}_m$ . Завдання декодера полягає у відновленні повідомлення  $m$  по відповідному округленому значенню  $\sigma = \hat{\sigma}_m$ . Це завдання вирішується за допомогою такого алгоритму.

### Алгоритм декодування коду Гілберта – Мура

1. Відомі  $q_m$ ,  $m = \overline{1, M}$ . Входом декодера служить число  $\hat{\sigma}$ , при цьому  $m = 1$ .
2. До тих пір, доки  $q_m < \hat{\sigma}$ , вважаємо  $m \leftarrow m + 1$ .
3. Результатом декодування є  $x_m$ .

Неважко перевірити, що алгоритм працює правильно. Для цього досить пригадати, що довжина кодових слів вибирається так, що в результаті округлення до  $l_m$  розрядів величини  $\sigma_m = q_m + p_m/2$  зменшуються не більше ніж на  $p_m/2$  (похибка округлення не більша ніж  $2^{-l_m} \leq p_m/2$ ). Тому мають місце нерівності

$$q_m \leq \hat{\sigma}_m \leq q_{m+1}.$$

У цьому алгоритмі при декодуванні використовувалися лише значення  $q_i$  та не використовувались значення  $\hat{\sigma}_i$ . Це важливо для застосування до декодування арифметичного коду. Так само при декодуванні арифметичного коду для отриманої з каналу округленої величини  $\hat{F}$  ми рекурентно обчислюємо найближче до  $\hat{F}$  значення, яке не перевищує  $\hat{F}$  зна-

чення кумулятивної ймовірності  $q(a)$ . Результатом декодування буде відповідна послідовність повідомлень  $X$ .

Розглянемо процес декодування. Декодеру відомі алфавіт  $A = \{1, \dots, M\}$ , ймовірність  $\{p_1, \dots, p_M\}$ , кумулятивна ймовірність  $\{q_1, \dots, q_M\}$ , довжина послідовності повідомлень  $n$  та отримане з каналу значення  $\hat{F}$ . Завдання полягає в обчисленні послідовності повідомлень  $A$ .

### Алгоритм декодування арифметичного коду

1. Ініціалізування початкових значень допоміжних змінних  $S = 0$ ,  $G = 1$ .
2. Для  $i$  від 1 до  $n$  виконуємо такі обчислення:
  - а)  $j = 1$ ;
  - б) доки  $S + q_{j+1}G < \hat{F}$ , вважаємо  $j \leftarrow j + 1$ ;
  - в) після завершення циклу отримуємо

$$S \leftarrow S + q_j G; \quad G \leftarrow p_j G; \quad a^{(i)} = j.$$

У цьому алгоритмі змінна  $G$  на  $i$ -му кроці дорівнює ймовірності  $p(a(i); 1)$  послідовності з перших  $i$  символів, а змінна  $S$  дорівнює кумулятивній ймовірності  $q(a(i); 1)$ .

*Приклад 3.4.* Розглянемо джерело  $A = \{a; b; c\}$ , в якому ймовірність  $p_a = 0,1$ ,  $p_b = 0,6$ ,  $p_c = 0,3$ . Припустимо, що на вхід декодера надійшла двійкова послідовність 0100010101, відповідно до якої необхідно відновити послідовність закодованих повідомлень. Простіше всього заглянути на декілька сторінок назад, адже в прикладі 3.3 ми одержали саме цю по-

слідовність на виході кодера. У табл. 3.7 наведені проміжні результати обчислень, виконаних при використанні описаного алгоритму декодування арифметичного коду. Як видно з табл. 3.7, послідовність повідомлень відновлена правильно.

Таблиця 3.7

**Декодування послідовності з прикладу 3.4**

Крок	$S$	$G$	Гіпотеза $a$	$q(a)$	$S + qG$	Рішення $a_i$	$p(a)$
$100010101 \rightarrow 0, 100010101 \rightarrow \hat{F} = 0,541$							
1	0,0000	1,0000	$a$	0,0	$0,0000 < \hat{F}$	$b$	0,6
			$b$	0,1	$0,1000 < \hat{F}$		
			$c$	0,7	$0,7000 > \hat{F}$		
2	0,1000	0,6000	$a$	0,0	$0,1000 < \hat{F}$	$c$	0,3
			$b$	0,1	$0,1600 < \hat{F}$		
			$c$	0,7	$0,5200 < \hat{F}$		
3	0,5200	0,1800	$a$	0,0	$0,5200 < \hat{F}$	$b$	0,6
			$b$	0,1	$0,5380 < \hat{F}$		
			$c$	0,7	$0,6460 > \hat{F}$		
4	0,5380	0,1080	$a$	0,0	$0,5380 < \hat{F}$	$a$	0,1
			$b$	0,1	$0,5488 > \hat{F}$		
5	0,5380	0,0108	$a$	0,0	$0,5380 < \hat{F}$	$b$	0,6
			$b$	0,1	$0,5391 < \hat{F}$		
			$c$	0,7	$0,5456 > \hat{F}$		

Повернемося до питання про обчислювальну складність. Суть проблеми полягає в тому, що у всіх обчисленнях як в кодері, так і декодері, ми виконуємо обчислення над змінними, розрядність яких дорівнює довжині закодованої послідовності повідомлень.

Розглянемо докладніше роботу кодера арифметичного коду при кодуванні послідовності з прикладу 3.3. Звернемося до табл. 3.6, з якої виходить, що після 3-го кроку кодування значення  $F$  уже не стане менше 0,5 і більше 0,75 незалежно від того, якими будуть подальші повідомлення джерела. Отже, перші два символи числа  $F$  уже не зміняться і можуть бути передані по каналу. Після цього можна виконати нормування  $F \leftarrow 2(F - 0,5)$ ,  $G \leftarrow 2G$  і продовжити кодування. Таке саме нормування виконає і декодер, у результаті чого розрядність змінних скоротиться без втрати в точності обчислень. Незавжди сформулювати загальне правило виконання такого роду нормувань.

Трудність виникає у тому випадку, коли двійкове подання  $F$  містить серію одиниць після нуля (число  $F$  близьке до 0,5, але менше 0,5), тобто  $F = 0,01111\dots$ . Невизначеність завершиться одним з двох способів. Або з'явиться нульовий розряд ( $F = 0,01111\dots 10$ ), або відбудеться перенесення в одному з молодших розрядів ( $F = 0,01111\dots 0$ ). Якщо стан невизначеності продовжиться довго, то розрядності комірок, використовуваних для зберігання  $F$  і  $G$ , може виявитися недостатньо.

Вихід з положення полягає в тому, що кодер, опинившись у такій ситуації, змінює форму зберігання числа  $F$ . Кількість розрядів зменшується за рахунок того, що в пам'яті зберігається не сама серія одиниць, а її довжина. У момент появи нуля на вихід видається послідовність вигляду  $0111 \dots 1$ . Якщо ж

невизначеність завершилася перенесенням, то на вихід надходить 1000 ... 0.

### 3.3. Методи кодування з урахуванням виявлення структурних закономірностей у зображеннях

*Структурне подання* зображення (даних) зводиться до подання форми складових його об'єктів і опису відношення між ними. Простими структурними формами є: відстані між найближчими позиціями ізотопного рівня; наявність перепадів між двома ізотопними рівнями; ширина ізотопного рівня або серії однакових елементів [3].

Для вивчення пропонуються перетворення початкових фрагментів зображень на основі виявлення простих структурних закономірностей, пов'язаних з описом ізотопних рівнів і топології колірних перепадів. На основі цього пропонується рівномірне структурне поелементне кодування без компресії. Сюди відносяться так звані інтервальні перетворення і перетворення за схемою “купка книг”. У цьому напрямку на розгляд пропонується блокове структурне рівномірне кодування довжин однакових елементів зображення, що називаються в літературі як довжини серій. Після чого буде показаний підхід для стиснення відеоданих на основі скорочення структурної надмірності, використовуючи методи сімейства LZ [29 – 32].

Розглянемо два способи взаємно однозначного перетворення послідовності букв (елементів) джерела в послідовність чисел натурального ряду. За певних умов (наприклад, при сильній залежності букв) поелементне кодування перетвореної послідовності виявляється помітно ефективнішим, ніж кодування початкової послідовності джерела.

До простих методів структурного рівномірного поелементного кодування без стиснення відносяться інтервальний код і код “купка книг”. Дамо трактування даним кодам з позиції структурного подання зображень, використовуючи для цього такі поняття, як топологія ізотопних рівнів та колірних перепадів. У зв'язку з цим, пропонуються такі трактування:

- інтервальний код – рівномірне кодування відстаней між двома найближчими позиціями одного ізотопного рівня;
- код “купка книг” – рівномірне кодування величин, які чисельно дорівнюють кількості колірних перепадів між двома найближчими позиціями одного ізотопного рівня.

*Кодування відстаней між найближчими позиціями ізотопного рівня (інтервальний код)*

Нехай  $\hat{A} = \{0, 1, \dots, M - 1\}$  – алфавіт джерела і на виході джерела спостерігається послідовність  $a_1, a_2, \dots$  (описується рекурентний алгоритм). Припустимо, що початкова частина послідовності  $a_1^{(n-1)} = (a_1, \dots, a_{n-1})$  уже закодована і передана і, отже, відома декодеру. Замість букви  $a_n$  передається довжина відстані (кількість букв) між попередньою і поточною появою даної букви. Іншими словами, кодер обчислює і передає мінімальне число  $d_n = d$  таке, що  $a_{n-r} = a_n$ . Тим самим початкова послідовність  $a_1, a_2, \dots$  перетвориться в послідовність чисел  $d_1, d_2, \dots$

Для завершення опису алгоритму потрібно визначити правило обчислення відстаней для тих букв, які не зустрічалися в послідовності  $a_1^{(n-1)}$ . Простіше всього умовитися про те, що першій передаваній букві передували всі букви алфавіту в

наперед узгодженому (для визначеності – в алфавітному) порядку.

*Приклад 3.5.* Є джерело з алфавітом  $A = \{a, b, c\}$ , яке породило послідовність *sabbbabbac*. Розглянемо роботу двох методів кодування в припущенні, що за умовчанням кодери (і декодери) вважають, що цій послідовності передувала послідовність  $\{a; b; c\}$ . Тоді при кодуванні відстаней між найближчими позиціями ізотопного рівня буде одержана послідовність  $\{0, 3, 3, 0, 0, 3, 1, 0, 2, 9\}$ .

*Кодування кількості колірних перепадів між найближчими позиціями одного ізотопного рівня (код “купка книг”)*

Цей метод має декілька назв і розкривається різними авторами. Вважається, що першим описав цей метод і проаналізував ступінь його ефективності Б. Я. Рябко [20]. Значно пізніше той же алгоритм у точності описаний під назвою “move-to-front coding”. Саме під такою назвою він найчастіше згадується в технічній літературі. У [18] пропонується інша назва цього методу “recency rank coding”. Це – остання назва, яку можна перекласти як “кодування ступеня новизни”.

Як і при кодуванні довжин інтервалів, припустимо, що початкова частина послідовності  $a_1^{(n-1)} = (a_1, \dots, a_{n-1})$  вже закодована та передана і, отже, відома декодеру. Але тепер замість букви  $a_n$  передається кількість різних букв між попередньою і поточною появою даної букви. Іншими словами, кодер обчислює та передає мінімальне число  $d_n = d$ , таке, що  $a_{n-r} = a_n$ . Потім обчислюється число  $q_n$  різних букв у послідовності  $a_{n-r+1}^{(n-1)}$ . Тим самим початкова послідовність  $a_1, a_2, \dots$  перетворюється в послідовність чисел  $q_1, q_2, \dots$

Для передачі тих букв, які не зустрічались у послідовності  $a_1^{(n-1)}$ , можна використовувати стратегію для кодування відстаней.

*Приклад 3.6.* Для умов попереднього прикладу 3.5 при кодуванні початкової послідовності одержимо послідовність  $\{0, 2, 2, 0, 0, 1, 1, 0, 1, 2\}$  за допомогою коду кількості кольорових перепадів.

Пояснимо походження назви “купка книг”. Припустимо, що студент, готуючись до іспиту, склав книги купкою на столі. Кожного разу, вибираючи з купки потрібну книгу, він не повертає її на своє місце, а кладе зверху. Верхня книга тепер має найменший номер, а номери книг, які раніше знаходились вище за неї, збільшилися на 1. Скільки книг знаходяться вище даної? Рівно стільки, скільки різних книг використовувалося між поточним та попереднім використанням даної книги. Таким чином, послідовність  $q_1, q_2, \dots$  якраз і є послідовністю номерів книг, які виймалися з купки.

Неважко зрозуміти, чому описані методи перетворення потоків даних можуть бути корисні при універсальному кодуванні. Чим більша ймовірність букви, тим коротше інтервали між її появами. Після перетворення послідовності джерела в послідовність інтервалів або послідовність номерів у купці книг найбільш імовірними будуть малі числа і менш імовірними великі. Тим самим створюються передумови для успішного застосування монотонних кодів. З опису двох розглянутих вище алгоритмів виходить, що з погляду застосування монотонних кодів метод “купки книг” має більше переваг, оскільки породжує числа, які менші або дорівнюють числам, породжуваним інтервальним кодером.

Отже, перетворення потоку даних за допомогою кодування відстаней або кодування за методом проміжної кількості кольорових перепадів з подальшим застосуванням коду Левенштейна або коду Елайсса гарантує достатньо ефективно універсальне кодування при дуже простій реалізації кодера та декодера. Ці методи не потребують накопичення статистичних даних про джерело і побудови відповідного коду джерела. Подібні ідеї лежать в основі алгоритмів Лемпеля – Зіва.

Прикладом блокових структурних кодів є рівномірне кодування довжин серій. *Довжина серії* – це лінійна ширина ізотопного рівня в рядковому напрямку, яка характеризує структурний зміст зображень. Тому виявлення довжин серій відноситься до методів структурного подання зображень. Довжини серій можуть називати в деяких літературних джерелах “довжини повторень”. За рахунок формування довжин серій забезпечуються умови для скорочення структурної надмірності зображень [3; 11]. Стиснення досягається, коли інформація про довжини серій відома заздалегідь, інакше, рівномірним кодуванням довжин серій не гарантується наявність компресії.

Ідея стиску даних на основі кодування довжин повторень полягає в тому, що замість кодування власне елементів зображень піддаються кодуванню числа, які відповідають довжинам ділянок, на яких дані зберігають незмінне значення.

Розглянемо приклад.

*Приклад 3.7.* Припустимо, що поточна послідовність елементів у рядку зображення виглядає таким чином:

75	75	75	75	75	75	75	75	75	60	60	60	53	61	61	80	80	80
80	80	79	81	78	75	75	75	75	75	75	75	75	75	75	75	75	75

Проскануємо цей фрагмент зображення по рядках, потім виділимо ділянки, на яких дані зберігають незмінне значення. У разі виявлення довжин серій початкова послідовність елементів зображення замінюється послідовністю

{(75;8);(60;3);(53;1);(61;2);(80;5);(79;1);(81;1);(78;1);(75;12)} .

Тепер цю послідовність, в якій помітна певна повторюваність (одиниць значно більше, ніж інших елементів), можна закодувати яким-небудь статистичним кодом, наприклад, кодом Хаффмана.

Нижче наведено інший приклад використання кодування довжин серій, коли в цифрових даних зустрічаються ділянки з великою кількістю нульових значень. Щоразу, якщо в потоці даних зустрічається “нуль”, він кодується двома числами (рис. 3.3).

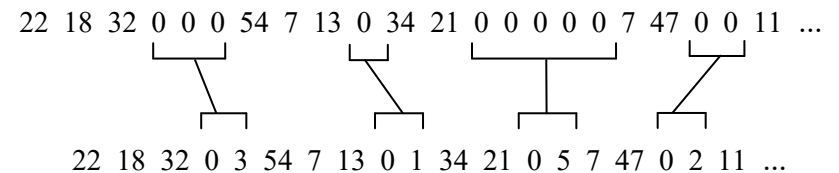


Рис. 3.3. Схема виявлення ланцюжків нульових елементів

Перше нульове значення є прапорцем початку кодування довжини потоку нулів, а друге – кількістю нулів у черговій групі. Якщо середня кількість нулів у групі більше двох, матиме місце стиснення. З іншого боку, велика кількість окремих нулів може призвести навіть до збільшення розміру кодованого файлу.

Кодування з виявленням довжин серій однакових елементів може використовуватися у таких випадках.

1. *Стиску початкових зображень (алгоритм Run Length Encoding – RLE)*. У цьому випадку початковими даними вважаються двовимірні масиви  $A = \{a_{ij}\}$ , елементи  $a_{ij}$  ( $i$  – індекс рядка,  $j$  – індекс стовпця), якими є оцифровані значення елементів зображення (пікселі). Суть даного підходу полягає в заміні ланцюгів або серій байтів, що повторюються, або їх послідовностей на один кодуєчий байт і лічильник кількості їх повторень. Проблема всіх аналогічних методів полягає лише у визначенні способу, за допомогою якого алгоритм, що розпаковує, міг би відрізнити в результуючому потоці байтів кодовану серію від інших некованих послідовностей байтів.

Вирішення проблеми досягається за допомогою застосування міток спочатку кодованих ланцюгів. Такими мітками можуть бути, наприклад, характерні значення біт у першому байті кодової серії, значення першого байта кодової серії і тому подібне.

Дані методи, як правило, достатньо ефективні для стиснення растрових графічних зображень (BMP, PCX, TIFF), компактного подання двійкових даних, наприклад, чорно-білих факсимільних зображень, чорно-білих зображень, що містять безліч прямих ліній і однорідних ділянок, схем. Такі зображення відрізняються змістом серій послідовностей байт, які повторюються.

Недоліком методу RLE є достатньо низький ступінь компресії або вартість кодування файлів з малою кількістю серій, а також з малою кількістю байтів, що повторюються, в серіях. До позитивних сторін алгоритму можна віднести, що він не потребує додаткової пам'яті при роботі і швидко виконується.

Цікава особливість групового кодування у форматі РСХ полягає в тому, що ступінь архівації для деяких зображень може бути істотно підвищений усього лише за рахунок зміни порядку кольорів у палітрі зображень.

2. *Стиск даних, отриманих на проміжних етапах процесу обробки (методи форматів JPEG, JPEG2000)*. Послідовностями даних можуть бути: кольороорієнтоване подання повнокольорових зображень; елементи диференційної імпульсно-кової модуляції подання; компоненти ортогональних перетворень; елементи бітової площини компонент ортогональних перетворень.

Загальна схема процесу стиснення зображень з виявленням довжин серій має вигляд, показаний на рис. 3.4.

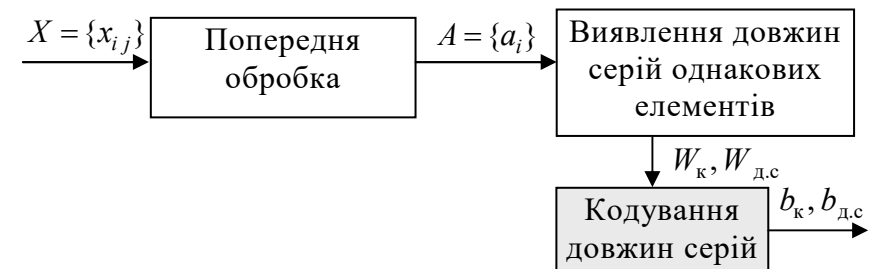


Рис. 3.4. Фрагмент схеми процесу стиснення з виявленням довжин серій

Послідовність даних  $A = \{a_i\}$  ( $i = \overline{1, Z}$ ,  $Z$  – довжина послідовності даних), яка отримана на попередньому етапі обробки (рис. 3.4) подається послідовністю  $\{c_\xi, l_\xi\}$  довжин серій  $l_\xi$  і значеннями  $c_\xi$  елемента серії ( $\xi = \overline{1, v_c}$ ,  $v_c$  – кількість серій, виявлених на послідовності даних). Якщо послідовність  $A$  є початковим зображенням, то  $c_\xi$  є значенням колірної ко-

ординати серії. Після формування довжин серій здійснюється їх кодування. Компактне подання даних з виявленням довжин серій полягає в кодуванні як колірних координат, так і довжин серій. Відповідно формуються кодові комбінації  $b_k$  та  $b_{д.с}$ . Для кодування довжин серій існують такі підходи [3].

1. Рівномірне кодування довжин серій. На подання кожної довжини серії відводиться однакова кількість двійкових розрядів  $b_\ell$ , яка дорівнює двійковому поданню заздалегідь встановленої максимальної довжини серії  $\ell_{\max}$ :

$$b_\ell = \log_2(\ell_{\max}) + 1.$$

Середній коефіцієнт  $k_{д.с}$  стиснення серії ( $\ell_\xi < \ell_{\max}$ ) визначається за формулою

$$k_{д.с} = \frac{b \sum_{\xi=1}^{v_c} \ell_\xi}{v_c b_\ell}.$$

Даний підхід характеризується відносно невеликою кількістю операцій на обробку. Найбільші ступені компактного подання досягаються при стисненні штучних зображень. Недоліки такого підходу виявляються при обробці реалістичних зображень, структурних властивостей (розподіл довжин серій наближається до рівномірного закону), що характеризуються нестационарністю, і невеликими серіями однакових елементів (1 – 4 елементи в серії). Це призводить до проблеми вибору максимальної довжини серії  $\ell_{\max}$ . З одного боку, для збіль-

шення ступеня стиску необхідно вибирати значення  $\ell_{\max}$  якомога більшим, але, з іншого боку, при кодуванні реалістичних зображень значення  $\ell_{\max}$  має бути заздалегідь меншим. Оскільки апіорі характеристики зображення невідомі, то неадекватний вибір  $\ell_{\max}$  призведе до зниження коефіцієнтів стиснення аж до збільшення первинного об'єму зображення.

2. Нерівномірне кодування довжин серій. Довжини серій, які мають велику ймовірність появи, подаються меншою кількістю двійкових розрядів і, навпаки, довжини серії, що рідше зустрічаються, кодуються більшою кількістю розрядів. У цьому випадку компресія послідовностей довжин серій досягається за рахунок скорочення статистичної надмірності. Найчастіше для кодування довжин серій використовуються різні варіанти арифметичного кодування. Арифметичне кодування забезпечує часткове вирішення проблеми вибору максимальної довжини серії. Проте арифметичне кодування характеризується істотними недоліками, що обмежують їх застосування для безпосередньої обробки відеоданих. До цих недоліків слід віднести таке:

– потенційно-максимальний ступінь стиснення  $k_{\max}$  довжин серій за рахунок зменшення статистичної надмірності залежить від ймовірності колірного перепаду (передбачається, що довжини серій мають геометричний розподіл). Для фрагментів, сильно насичених дрібними деталями, ймовірності колірного перепаду  $p > 0,2$ . Це призводить до зниження потенційних можливостей арифметичного кодування щодо усунення статистичної надмірності і до збільшення похибки при оцінці ймовірності появи значень довжин серій;

– не враховуються двовимірні закономірності для структурних властивостей зображень;

– при збільшенні динамічного діапазону оброблюваних даних відбувається збільшення значення нижньої межі оцінки кількості інформації;

– арифметичним кодовим комбінаціям при їх передачі по каналах зв'язку з помилками властива низька завадостійкість. Спотворення навіть одного розряду призводить до спотворення всіх невідновлених на даний момент елементів. Уведення розділювачів призводить до зниження ступеня стиску. У той же час довжини серій також мають низьку завадостійкість до помилок у каналі зв'язку. Помилка в одній довжині серії призведе до зсуву рядків зображень при відновленні подальших довжин серій. Отже, арифметичне кодування довжин серій додатково збільшить ступінь впливу помилок у каналі зв'язку на якість відновленого зображення. При наявності помилки в одному кодовому слові не можна відновити все повідомлення.

У найбільш сприятливих умовах розподіл імовірності появи довжин серій описується геометричним законом розподілу. У цьому випадку для кодування довжин серій більші переваги порівняно з рівномірним має монотонне кодування.

### **Методи структурного кодування сімейства LZ (метод Лемпеля – Зіва)**

Такі методи називають підстановочними або словарно-орієнтованими методами стиснення [26; 29 – 32].

Практично всі словарні методи кодування належать класу алгоритмів, наведених у роботі двох ізраїльських вчених – Лемпеля (Lempel) і Зіва (Ziv). Історично першим був алгоритм

LZ77, опублікований в 1977 році та названий так на честь авторів. У багатьох програмах компресії інформації використовуються ті або інші модифікації LZ77. Однією з причин популярності алгоритмів LZ є їх виняткова простота при високій ефективності стиснення.

Суть їх полягає в тому, що фрази тексту, що стискається, замінюються покажчиком на координату тієї позиції, в якій вони в цьому тексті з'являлись раніше.

Методи LZ-сімейства швидко пристосовуються до структури тексту та можуть кодувати короткі функціональні слова, які дуже часто в ньому з'являються. Нові слова і фрази можуть також формуватися з частин слів, які зустрічались раніше.

Декодування компактно поданого тексту здійснюється безпосередньо, тобто відбувається проста заміна покажчика готовою фразою зі словника, на яку той указує. На практиці за допомогою LZ-методу добиваються хорошого стиснення. Цей метод відрізняється дуже швидкою роботою декодера. (Під текстом розуміється деякий вектор даних з кінцевим дискретним алфавітом, і це не обов'язково текст у буквальному розумінні цього слова).

Більшість словарних методів кодування має ім'я авторів ідеї методу Лемпеля і Зіва, але не підпідставно часто вважається, що у всіх цих методах використовується один і той же алгоритм кодування. Насправді представники цього сімейства алгоритмів дуже сильно різняться деталями своєї роботи. При цьому всі словарні методи кодування можна розбити на дві групи (табл. 3.8).

Таблиця. 3.8

**Хронологічна класифікація методів LZ-сімейства**

Дата публікації	LZ-сімейство	
1977 р.	LZ77	
1978 р.		LZ78
1982 р.	LZSS	
1984 р.		LZW

За допомогою методів, що належать до першої групи, відшукуючи в кодованій послідовності ланцюги символів, які раніше вже зустрічались, не повторюють ці ланцюги, а замінюють їх покажчиками на попередні повторення.

Словник цієї групи алгоритмів у неявному вигляді міститься в оброблюваних даних, а зберігаються лише покажчики на ланцюги символів, що повторюються.

Усі методи цієї групи базуються на алгоритмі LZ77.

Алгоритми другої групи на доповнення до початкового словника джерела створюють словник фраз, що є комбінаціями символів початкового словника, які повторно зустрічаються у вхідних даних. При цьому розмір словника джерела зростає, і для його кодування буде потрібно більше число біт, але значна частина цього словника буде вже не окремими буквами, а цілими словами. Коли кодер виявляє фразу, яка раніше вже зустрічалась, він замінює її індексом словника, що містить цю фразу. При цьому довжина коду індексу виходить меншою або набагато меншою ніж довжина коду фрази.

Усі методи цієї групи базуються на алгоритмі LZ78, розробленому та опублікованому Лемпелом і Зівом у 1978 році.

Алгоритми LZ77, LZ78 та LZSS розроблені математиками і можуть використовуватися вільно.

**Алгоритми LZ77 та LZSS**

Основна ідея LZ77 полягає в тому, що друге і подальші входження деякого рядка символів у повідомленні замінюються посиланнями на їх перше входження. Метод LZ77 використовує вже проглянуту частину повідомлення як словник. Щоб добитися стиснення, проводиться заміна чергового фрагмента повідомлення на покажчик у змісті словника. Алгоритм LZ77 використовує вікно, що «ковзає» по повідомленню, розділене на дві нерівні частини. Перша, велика за розміром, включає в себе вже проглянуту частину повідомлення. Друга, набагато менша, є буфером, що містить ще незакодовані символи вхідного потоку.

Зазвичай розмір вікна становить декілька кілобайт, а розмір буфера – не більше ста байт. Алгоритм намагається знайти в словнику (більшій частині вікна) фрагмент, який збігається з вмістом буфера.

Алгоритм LZ77 видає коди, до складу яких входить три елементи, а саме:

- зсув у словнику відносно початку його підрядка, який збігається з початком вмісту буфера;
- довжина цього підрядка;
- перший символ буфера, який надходить за підрядком.

Декодування коду LZ77 простіше за його отримання, оскільки не потребує здійснення пошуку в словнику. Недоліки методу LZ77 такі:

1) зі зростанням розмірів словника швидкість роботи алгоритму-кодера пропорційно сповільнюється;

2) кодування одиночних символів дуже неефективне.

Довжина коду обчислюється таким чином: довжина підрядка не може бути більше розміру буфера, а зсув – більше розміру словника, зменшеного на одиницю. Отже, довжина двійкового коду зсуву буде округленою у більший бік  $\log_2$  (розмір словника), довжина двійкового коду для довжини підрядка буде округленою у більший бік  $\log_2$  (розмір буфера + 1), а символ кодується 8-ма бітами (згідно з кодовим поданням елемента колірної площини зображення).

Процедура кодування відповідно до алгоритмів цієї групи ілюструється рис. 3.5 [26].

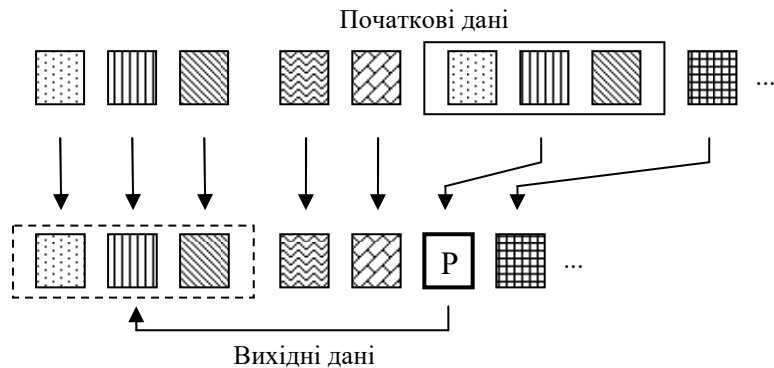


Рис. 3.5. Схема алгоритмів типу LZ77

Розглянемо формальний опис алгоритму.

Параметром алгоритму є довжина “вікна спостереження”  $W$ . Цю величину можна також інтерпретувати як “обсяг змінного словника”.

Будемо вважати, що  $A = \{0, 1, \dots, M - 1\}$  – алфавіт джерела та на виході джерела спостерігається послідовність  $a_1, a_2, \dots, a_m$ . Алгоритм описується рекурентно. Припустимо, що початкова частина послідовності  $a_1^{(n)} = (a_1, \dots, a_n)$  вже закодована та передана, отже, відома декодеру. Виконуються наступні обчислення.

**Алгоритм LZ77** передбачає виконання таких операцій.

1. Визначається максимальне значення  $l$ , таке, за якого для деякого  $d$ ,  $d \in \{1, \dots, W\}$  послідовність  $a_{n+1}^{(n+l)}$  (тобто послідовність із  $l$  наступних символів) збігається з деякою послідовністю  $a_{n-d+1}^{(n-d+l)}$  довжини  $l$ , що спостерігалася в послідовності  $a_{n-W+1}^{(n)}$ .

2. Якщо  $l > 0$ , то до кодової послідовності дописується:

– символ 1, що вказує на існування нетривіальної послідовності  $a_{n+1}^{(n+l)} = a_{n-d+1}^{(n-d+l)}$ ,  $d \in \{1, \dots, W\}$ ;

– послідовність із  $\lceil \log W \rceil$  біт, що являє собою двійковий запис числа  $d$ ;

– число  $l$ , що записане у вигляді кодового слова деякого нерівномірного префіксного коду.

При цьому значення  $n$  збільшується на довжину  $l$  та кодер переходить до кодування наступних символів джерела (крок 1).

У протилежному разі ( $l = 0$ ) передається:

– символ 0, який вказує на те, що буква  $a_{n+1}$  не зустрічалася у вікні  $a_{n-W+1}^{(n)}$ ;

– послідовність із  $\lceil \log M \rceil$  біт, що являє собою двійковий запис номера букви  $a_{n+1}$  в алфавіті  $A$ .

При цьому значення  $n$  збільшується на 1 і кодер переходить до кодування наступних символів джерела (крок 1).

Зазначимо, що на початку роботи алгоритму, тобто при  $n < W$ , фактична довжина вікна дорівнює  $n$  та для передачі числа  $d$  достатньо  $\lceil \log n \rceil$  біт (замість  $\lceil \log W \rceil$  біт).

Отже, кодер алгоритму LZ77 зберігає в пам'яті змінний словник обсягу  $W$ . Словами словника служать підпослідовності букв, що надходять одна за одною та містяться в останніх  $W$  буквах джерела. При надходженні на вхід кодера нових букв кодер знаходить якомога довшу послідовність, уже наявну в словнику. У канал передається прапорець (1 або 0), який визначає, чи знайдене підходяще словникове слово. У разі успіху (прапорець дорівнює 1) словникове слово передається указанням видалення початку слова від поточної позиції та довжини словникового слова. Відстань до слова  $d$  передається рівномірним кодом, а довжина слова – деяким нерівномірним кодом. Наприклад, можна скористатися кожним з монотонних кодів. Якщо ж словникового слова не знайшлося, то передається значення прапорця, що дорівнює 0 і за ним надходить чергова буква джерела, передана “без кодування”.

Пояснимо роботу цього нескладного алгоритму прикладом.

*Приклад 3.8.* Застосуємо алгоритм LZ77 до прислів'я

*IF WE CANNOT DO AS WE WOULD WE SHOULD DO AS WE CAN*

Припустимо, що довжини слів  $l$  кодуються монотонним кодом (3.2). Результати роботи алгоритму крок за кроком ві-

дображені в табл. 3.9. У цій таблиці  $\text{bin}(\cdot)$  означає стандартний 8-бітний код букви, у графі “відстань” у дужках зазначена поточна довжина вікна. Від неї залежить кількість бітів, що відводяться на передачу відстані між поточною позицією та знайденим у табл. 3.9 зразком. Сумарні витрати на передачу послідовності становлять 257 бітів.

З прикладу виходить, що алгоритм LZ77 істотно виграє по стисненню порівняно з логічно більш складними методами адаптивного арифметичного кодування.

Більш того, з табл. 3.9 ясно видно, що ефективність алгоритму швидко зростає при збільшенні обсягу змінного словника.

*Таблиця 3.9*

**Метод змінного словника (LZ77)**

Крок	Прапорець	Послідовність символів	Відстань $d$	Довжина $l$	Кодова послідовність	Витрати (біт)
0	0	$I$	–	0	0 $\text{bin}(I)=0$	9
1	0	$F$	–	0	0 $\text{bin}(F)$	9
2	0	_	–	0	0 $\text{bin}(\_)$	9
3	0	$W$	–	0	0 $\text{bin}(W)$	9
4	0	$E$	–	0	0 $\text{bin}(E)$	9
5	1	_	2(5)	1	1 010 0	5
6	0	$C$	–	0	0 $\text{bin}(C)$	9
7	0	$A$	–	0	0 $\text{bin}(A)$	9
8	0	$N$	–	0	0 $\text{bin}(N)$	9

Закінчення табл. 3.9

Крок	Пра- порець	Послідов- ність сим- волів	Відстань $d$	Дов- жина $l$	Кодова послідовність	Вит- рати (біт)
9	1	$N$	0(9)	1	1 0000 0	6
10	0	$O$	–	0	0 bin( $O$ )	9
11	0	$T$	–	0	0 bin( $T$ )	9
12	1	_	6(12)	1	1 0110 0	6
13	0	$D$	–	0	0 bin( $D$ )	9
14	1	$O$	3(14)	1	1 0011 0	6
15	1	_	2(15)	1	1 0010 0	6
16	1	$A$	8(16)	1	1 00100 0	7
17	0	$S$	–	0	0bin( $S$ )	9
18	1	_WE_	15(18)	4	1 01111 11000	11
19	1	$W$	2(22)	1	1 00010 0	7
20	1	$O$	8(23)	1	1 00100 0	7
21	0	$U$	–	0	0 bin( $U$ )	9
22	0	$L$	–	0	0 bin( $L$ )	9
23	1	$D$	12(26)	1	1 01100 0	7
24	1	_WE_	8(27)	4	1 00100 11000	11
25	1	$S$	13(28)	1	1 011010	7
26	0	$H$	–	0	0 bin( $H$ )	9
27	1	OULD_	9(33)	5	1 001001 11001	12
28	1	DO_AS_ WE	24(38)	9	1 011000 1110001	14
29	1	CAN	40(47)	3	1 101000 101	10
Разом						257

Наступні міркування дозволяють зрозуміти характер по-  
водження алгоритму зі збільшенням обсягу словника. Дійсно,

довжина кодової послідовності, що зіставляється з відрізком  
 $a_{n+1}^{(n+l)}$ , становить

$$L(a_{n+1}^{(n+l)}) \approx 1 + \log(W) + 2 \log l \text{ [біт]}.$$

Ці біти витрачені на передачу  $l$  букв джерела. Значимо,  
що зі збільшенням обсягу словника збільшуються довжини  
послідовностей, які кодер знаходить у словнику. Таким чином,  
при великих значеннях  $W$  (і відповідно  $l$ ) “миттєва швид-  
кість” кодування приблизно дорівнює

$$R(a_{n+1}^{(n+l)}) \approx \frac{1 + \log W + 2 \log l}{l} = \frac{\log W}{l} + o(l),$$

де  $o(l) \rightarrow 0$  при збільшенні  $W$  та  $l$ .

Тепер потрібно встановити зв'язок між  $W$  і  $l$  при великих  
 $W$ . Зрозуміло, що сполучення букв, яке має ймовірність  $p$ , зу-  
стріється в словнику обсягом  $W$  приблизно  $pW$  раз. Якщо  
 $pW \gg 1$ , то, найімовірніше, сполучення букв має продовжен-  
ня, яке також є в словнику. Звідси виходить, що типові пого-  
джені послідовності спостерігаються у вікні довжиною  $W$  кін-  
цеву кількість разів і що при великих значеннях  $W$  має місце на-  
ближене співвідношення

$$p(a_{n+1}^{(n+l)}) \approx \frac{K}{W},$$

де  $K$  – константа.

Підставивши це співвідношення у вирази для швидкості  
кодування, одержимо

$$R(a_{n+1}^{(n+l)}) \approx \frac{1}{l} \log \frac{1}{p(a_{n+1}^{(n+l)})} + \frac{\log K}{l} = o(l).$$

Усреднюючи по множині послідовностей джерела, доходимо висновку, що

$$\bar{R} = M[R(a_{n+1}^{(n+l)})] \xrightarrow{l \rightarrow \infty} H(A)_\infty.$$

Можна оцінити швидкість спадання надмірності зі збільшенням  $W$ . Основний внесок у надмірність вносить доповнення, що пов'язане з необхідністю передачі довжини погодженого відрізка, тобто надмірність має порядок  $(\log l)/l$ . Оскільки асимптотично (для типових послідовностей)  $W$  пропорційно  $\log l$ , то асимптотичне поведіння надмірності кодування зі збільшенням  $W$  визначається формулою

$$\bar{R} - H(A)_\infty \approx \frac{\log \log W}{\log W}.$$

Швидкість спадання надмірності універсального кодування послідовності довжини  $n$  для методів адаптивного арифметичного кодування має порядок  $\log n/n$ . Це означає, що надмірність алгоритму LZ77 спадає (асимптотично) набагато повільніше. Проте співвідношення між складністю алгоритму та стисненням саме для цього алгоритму виявилось найкращим серед відомих на сьогодні алгоритмів.

Розглянемо коротко деякі аспекти практичної реалізації алгоритму і його основні модифікації.

Обговоримо складність алгоритму. Насамперед, очевидно, складності кодування та декодування не рівні. Обчислювальна складність декодування дуже низька. Декодер безпосередньо з бітового потоку одержує інформацію про те, починаючи з якої букви і якої довжини послідовність букв він повинен витягти з пам'яті та видати одержувачу.

Навпаки, кодер для кожного зі значень  $l = 1, 2, \dots$  має повторити спробу пошуку у вікні довжини  $W$  (типові розміри вікна становлять 2 048 – 16 384 символів) зразків, які збігаються з буквами джерела, що надійшли знову. Така “прямо-лінійна” реалізація алгоритму неприйнятно складна. Застосовувані в архіваторах кодери для швидкого пошуку зразків у словнику кодера використовують хешування.

Розглянемо докладніше алгоритм роботи швидкого кодера, що використовує алгоритм LZ77. Насамперед, зазначимо, що зі збільшенням довжини вікна посилання на послідовності з 1 або 2 символів стають неефективними. Такі послідовності простіше передати по одній букві, затрачаючи по 9 біт на букву, як це робиться для букв, що зустрілися знову. Трибуквені послідовності розглядаються як двійкові послідовності довжини 24, і для них обчислюється хеш-функція, що відображає послідовність довжини 24 у двійкову послідовність довжини 12. Ця послідовність використовується як адреса комірки пам'яті, у якій зберігається указання на місце у вікні, де останній раз зустрічалося таке ж сполучення букв. Ця інформація використовується далі при кодуванні, а на її місце записується указання на поточну позицію.

Кодер, маючи інформацію про останню появу трибуквених сполучень, перевіряє дійсну довжину збігу. Далі кодер має перевіряти інші місця у вікні, де були такі ж сполучення букв. У багатьох завданнях, особливо, при реалізації кодерів на великих інтегральних схемах, це цілком розумна плата за істотне спрощення та прискорення роботи кодера. Якщо ж потрібне більш високе стиснення, необхідно для кожної позиції вікна зберігати та оновлювати інформацію про те, в якій точці вікна востаннє зустрілося таке ж сполучення букв, як і в поточній позиції.

З урахуванням цих спрощень можна вважати алгоритм LZ77 на сьогодні найшвидшим і що на основі його модифікацій побудовані практично всі архіватори.

Розглянемо тепер підходи до підвищення ефективності кодування. Нагадаємо, що на кожному кроці кодування формуються 3 випадкові величини: двійковий прапорець, посилання, що приймає значення з кінцевого діапазону  $\{1, \dots, W\}$ , і довжина посилання, яка теоретично може бути будь-яким натуральним числом, але на практиці великі значення довжин зустрічаються рідко.

**Першим** шляхом до поліпшення кодування є використання більш ефективних кодів для цих величин, ніж розглянуті вище. Зокрема, можна використовувати адаптивне кодування на основі арифметичного кодування або кодів Хаффмана. Ця можливість використовується у всіх архіваторах.

Показовою в цьому сенсі є модифікація LZ77, запропонована Фіалом і Гріном [18]. Її називають алгоритмом LZFG. Розглянемо спочатку найпростіший варіант алгоритму LZFG.

Як і в LZ77, на кожному кроці кодування формуються кодові комбінації двох типів, один з них відповідає посиланню

на послідовності букв, що вже містяться у вікні, інший – безпосередній передачі послідовності букв джерела.

**Друга** можливість збільшення стиснення полягає в усуненні надмірності кодування, що свідомо властива даному алгоритму. Оскільки в змінному словнику є однакові послідовності однієї й тієї послідовності букв джерела, якій можуть бути зіставлені різні кодові слова, то деякі кодові слова просто ніколи не використовуються.

У роботі [18] описаний спосіб зберігання словника у вигляді деревоподібної структури. Ребрам дерева відповідають послідовності букв. Посилання дозволені тільки на вузли дерева. Оскільки кількість вузлів менша загальної кількості букв, то зменшуються витрати на передачу адреси слова в словнику.

У різних статтях і патентах можна знайти багато інших методів удосконалення алгоритму LZ77.

У 1982 р. Сторером (Storer) і Шиманським (Szimanski) на базі методу LZ77 було розроблено алгоритм LZSS, який відрізняється від LZ77 вироблюваними кодами. Даний алгоритм є найбільш досконалим представником цієї групи методів, що увібрав у себе всі досягнення, отримані в даному напрямку.

Код, який формується алгоритмом LZSS, починається з однобітового префікса, що відрізняє код від незакодованого символу. Код складається зі зсуву та довжини, таких же, як і для методу LZ77. У методі LZSS вікно зсувається рівно на довжину знайденого підрядка або на 1, якщо не знайдено входу підрядка з буфера в словник. Довжина підрядка методу LZSS завжди більше нуля, тому довжина двійкового коду для довжини підрядка є округленим до більшого цілого двійкового логарифма від довжини буфера.

Алгоритми LZ77 та LZSS мають такі недоліки:

1) неможливість кодування підрядків, віддалених один від одного на відстань, яка більше ніж довжина словника;

2) довжина підрядка, який можна закодувати, обмежена розміром буфера.

Якщо надмірно збільшувати розміри словника та буфера, то це призведе до зниження ефективності кодування, оскільки зі зростанням цих величин ростимуть і довжина коду для зсуву і довжина коду підрядка, що зробить коди для коротких підрядків неприпустимо великими. Крім того, різко збільшиться час роботи алгоритму-кодера.

### Алгоритми LZ78 та LZW

Метод LZ78 не використовує “ковзаюче” вікно; він зберігає словник із вже проглянутих фраз. При старті алгоритму цей словник містить тільки один вільний рядок (рядок довжини нуль). Алгоритм зчитує символи повідомлення до тих пір, доки накопичуваний підрядок входить повністю в одну з фраз словника. Як тільки цей рядок перестане відповідати хоча б одній фразі словника, алгоритм генерує код, що складається з індексу рядка в словнику, який до останнього введеного символу містив вхідний рядок, і символу, що порушив збіг. Потім у словник додається введений підрядок. Якщо словник уже заповнено, то з нього заздалегідь видаляють найменш використовувану в порівняннях фразу.

Ключовим для отримуваних кодів є розмір словника у фразах, тому що кожен код при кодуванні за методом LZ78 містить номер фрази в словнику. З останнього виходить, що ці коди мають постійну довжину, яка дорівнює округленому у більший бік двійковому логарифму розміру словника +8 (це

кількість біт для подання елемента колірної площини зображення).

Ідею цієї групи алгоритмів можна також пояснити за допомогою рис. 3.6 [18; 26].

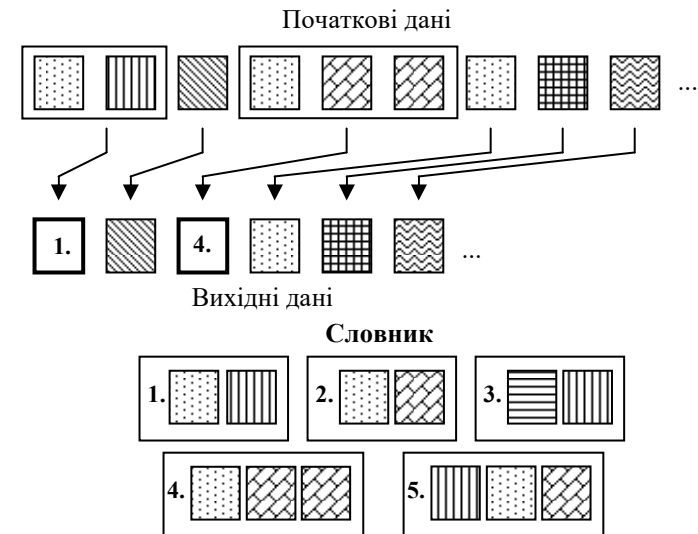


Рис. 3.6. Схема підстановок в алгоритмах типу LZ78

У 1984 р. Тері Уелчем (Welch) шляхом модифікації методу LZ78 було створено алгоритм LZW. На даний момент метод LZW є найбільш досконалим представником цієї групи словарних методів.

Розглянемо в найзагальнішому вигляді роботу LZW-кодера та декодера.

У процесі роботи алгоритму кодер і декодер синхронно формують словник. Цей словник на кожному кроці поповнюється одним новим словом, яке до цього в словнику було відсутнє, але є продовженням на одну букву одного зі слів словника.

Будемо вважати, що  $A = \{0, 1, \dots, M - 1\}$  – алфавіт джерела та на виході джерела спостерігається послідовність  $a_1, a_2, \dots$ . Алгоритм описується рекурентно. Припустимо, що початкова частина послідовності  $a_1^{(n)} = (a_1, \dots, a_n)$  вже закодована та передана, а також, що при цьому (і кодером і декодером) побудовано однаковий словник з  $c$  слів. Словник утворений словами різної довжини з букв алфавіту  $A$ . Для простоти опису алгоритму будемо вважати, що кожна з букв алфавіту є словом довжини  $l$  і входить до складу словника. (Усі букви можуть бути записані в словник до початку роботи алгоритму). На черговому кроці алгоритму виконуються такі обчислення.

Знаходимо максимальне  $l$  таке, що для деякого  $j$ ,  $j \in \{1, \dots, c - 1\}$  послідовність  $a_{n+1}^{(n+l)}$  (тобто послідовність із  $l$  наступних символів) збігається з  $j$ -м словом словника. До кодової послідовності дописується двійкова послідовність довжини  $\lceil \log(c - 1) \rceil$ , що являє собою двійковий запис номера словникового слова  $j$ . У словник дописується нове слово довжини  $l + 1$  виду  $a_{n+1}^{(n+l+1)}$ . Кількість  $n$  збільшується на  $l$ , обсяг словника  $c$  збільшується на 1. Кодер переходить до кодування наступних символів джерела.

Значимо, що знову дописане в словник слово складається з уже наявного в словнику слова  $a_{n+1}^{(n+l)}$  та наступної за ним букви джерела  $a_{n+l+1}$ . Слова  $a_{n+1}^{(n+l+1)} = (a_{n+1}^{(n+l)}, a_{n+l+1})$  у словнику немає, оскільки інакше довжина збігу  $l$  була б на одиницю більше, що суперечить алгоритму. Декодер одержить із каналу тільки указання на словникове слово  $a_{n+1}^{(n+l)}$ . Цього зазвичай

недостатньо для побудови нового  $(c + 1)$ -го слова  $a_{n+1}^{(n+l+1)}$ , але відсутня буква  $a_{n+l+1}$  буде першою буквою словникового слова, знайденого на наступному кроці кодування. Таким чином, декодер буде будувати словник із запізненням на один крок. Саме тому в описі алгоритму дозволяється вибирати слова з номерами, не більше  $c - 1$ , і на передачу номера відводимо  $\lceil \log(c - 1) \rceil$  біт.

Ще одне зауваження стосується ініціалізації алгоритму. При кодуванні коротких послідовностей недоцільно ініціалізувати алгоритм, заповнюючи словник усіма буквами алфавіту джерела. Доцільно використовувати принцип esc-коду. На початку роботи алгоритму словник порожній і в ньому в комірці з нульовим номером зберігається esc-код. Надалі, при одержанні від джерела букви, яка відсутня в словнику, передається esc-код, якому відповідає нульова послідовність довжини  $\lceil \log(c - 1) \rceil$ , а слідом за esc-кодом – стандартний код нової букви.

Робота кодера пояснюється таким прикладом.

*Приклад 3.9.* Застосуємо алгоритм LZW до прислів'я

*IF\_WE\_CANNOT\_DO\_AS\_WE\_WOULD\_WE\_SHOULD\_DO\_AS\_WE\_CAN*

Результати роботи алгоритму по кроках наведені в табл. 3.10.

Витрати на кодування в даному прикладі виявилися вище, ніж для інших способів. Але відносно моменту закінчення кодування слід зазначити, що словник буде містити корисні сполучення букв, які можна застосовувати при кодуванні подальшого тексту.

Таблиця 3.10

## Приклад використання алгоритму LZW

Крок	Словник	Номер слова	Кодова послідовність	Кількість біт
0	esc	–	–	–
1	<i>I</i>	0	bin( <i>I</i> )	8=8
2	<i>F</i>	0	bin( <i>F</i> )	log(1)+8=8
3	–	0	0bin( )	$\lceil \log(2) \rceil + 8 = 9$
4	<i>W</i>	0	00 bin( <i>W</i> )	$\lceil \log(3) \rceil + 8 = 10$
5	<i>E</i>	0	00 bin( <i>E</i> )	$\lceil \log(4) \rceil + 8 = 10$
6	<i>_C</i>	3	011	3
7	<i>C</i>	0	000bin( <i>C</i> )	3+8=11
8	<i>A</i>	0	000bin( <i>A</i> )	3+8=11
9	<i>N</i>	0	000bin( <i>N</i> )	3+8=11
10	<i>N</i>	0	000bin( <i>N</i> )	4+8=12
11	<i>O</i>	0	0000bin( <i>O</i> )	4+8=12
12	<i>T</i>	0	0000bin( <i>T</i> )	4+8=12
13	<i>_D</i>	3	0011	4
14	<i>D</i>	0	0000bin( <i>D</i> )	4+8=12
15	<i>O_</i>	11	1011	4=4
16	<i>_A</i>	3	0011	4
17	<i>AS</i>	8	1000	4
18	<i>S</i>	0	00000bin( <i>S</i> )	5+8
19	<i>_W</i>	3	00011	5
20	<i>WE</i>	4	00100	5
21	<i>E_</i>	5	00101	5
22	<i>_WO</i>	19	10011	5
23	<i>OU</i>	3	00011	5

Закінчення табл. 3.10

Крок	Словник	Номер слова	Кодова послідовність	Кількість біт
24	<i>U</i>	0	00000bin( <i>U</i> )	5+8=13
25	<i>L</i>	0	00000bin( <i>L</i> )	5+8=13
26	<i>D_</i>	14	01110	5
27	<i>_WE</i>	19	10011	5
28	<i>E_S</i>	21	10101	5
29	<i>SH</i>	18	10010	5
30	<i>H</i>	0	00000bin( <i>H</i> )	5+8=13
31	<i>OUL</i>	23	10111	5
32	<i>LD</i>	25	11001	5
33	<i>D_D</i>	26	11010	5
34	<i>DO</i>	14	001110	6
35	<i>O_A</i>	15	001111	6
36	<i>AS_</i>	17	010001	6
37	<i>_WE_</i>	27	011011	6
38	<i>_CA</i>	6	000110	6
39	<i>AN</i>	8	001000	6
40	<i>N</i>	9	001001	6
Разом				299

Таким чином, очікується, що зі збільшенням довжини послідовності, що кодується, ефективність кодування буде швидко наростати.

З погляду практичного застосування алгоритму LZW важливо відповісти на два питання: як організувати зберігання в пам'яті кодера та декодера великого словника, що складений зі слів різної довжини, і як діяти при заповненні цим словником усього припустимого обсягу пам'яті.

Відповідь на перше питання міститься в роботі Велча [29]. Оскільки словник має деревоподібну структуру, то й зберігати його потрібно у вигляді дерева, передаючи для кожного вузла дерева адреси сусідньої вершини та адреси старшого нащадка вершини. При такій структурі пошук у словнику також прискорюється. Крім цього, додаткове прискорення пошуку можливо за допомогою хешування.

Друге питання складніше. Тривіальними вирішеннями проблеми є:

- припинення нарощування словника після того, як вичерпана припустима пам'ять;
- повне стирання пам'яті та повернення алгоритму до початкового стану.

Таким чином слід зазначити, що базові алгоритми кодування дозволяють створювати потужні методи стиснення цифрових зображень повітряної розвідки.

### ***Контрольні питання та завдання***

1. Дайте визначення такого поняття, як кодування.
2. Що таке поелементний код Бодо?
3. Назвіть особливості формування узагальненого коду Бодо.
4. Дайте визначення монотонних кодів.
5. Назвіть ознаки префіксних нерівномірних кодів.
6. Чи впливає вимога префіксності на оптимальність кодування?
7. Яку структуру має код Голомба – Райса?
8. Яку структуру має код Левенштейна?

9. За якими ознаками проводять розділення джерел інформації на класи в теорії інформації?

10. Як математично описується джерело Хартлі?

11. Які властивості мають повідомлення джерела Бернуллі і в чому відмінність цього типу джерела від джерела Хартлі?

12. Як проводиться математичний опис джерела Бернуллі?

13. Які джерела називають джерелами Маркова, в чому їх відмінність від джерел Хартлі і Бернуллі?

14. Як математично описуються джерела Маркова?

15. У зв'язку з чим виникає необхідність моделювати джерела дискретних повідомлень двозв'язковими послідовностями Маркова?

16. Що характеризує собою величина умовної ентропії двозв'язкового дискретного джерела?

17. Як обчислюється величина інформаційної продуктивності двозв'язкового марковського дискретного джерела?

18. Як обчислити величину інформаційної продуктивності багатозв'язкових марковських дискретних джерел?

19. У яких відношеннях знаходяться значення інформаційної продуктивності повідомлень дискретних джерел при збільшенні міри зв'язності символів у повідомленнях?

20. Назвіть основні етапи алгоритму Хаффмана.

21. Чому алгоритм Хаффмана вважається оптимальним для статистичних джерел?

22. Назвіть недоліки методу кодування Хаффмана.

23. Дайте визначення кодів з пам'яттю.

24. Які існують реалізації кодів з пам'яттю?

25. Назвіть особливості побудови арифметичного коду.

26. Які особливості має практична реалізація арифметичного коду?
27. Назвіть основні етапи декодування арифметичних кодів.
28. Які особливості має структурний підхід відносно стиснення даних?
29. Назвіть основні структурні перетворення послідовностей.
30. У чому суть кодів довжин серій?
31. Назвіть основні особливості та суть методів сімейства LZ.
32. Назвіть основні характерні риси методів групи LZ77.
33. Назвіть основні характерні риси методів групи LZ78.
34. Назвіть основні етапи методу LZW.

## Л І Т Е Р А Т У Р А

### Розділ 1, 2

1. Бараннік В. В. Методи та засоби формування і обробки видових зображень в системі Повітряної розвідки Гриф МОН України: навч. посіб. / С. А. Сідченко, Б. М. Іващук – Х.: ХУПС, 2012. – 568 с.
2. Наукоємні технології в інфокомунікаціях: обробка, захист та передача інформації: монографія / В. М. Безрук, В. В. Бараннік. – Х.: ХНУРЕ, ФОП Бровін О. В., 2018. – 328 с.
3. Методы повышения информационной безопасности в системах видеомониторинга кризисных ситуаций: монографія / В. В. Бараннік, Ю. М. Рябуха. – Ч.: ЧППБ ім. Героїв Чорнобиля НУЦЗ України, 2015. – 143 с.
4. Бондарев В. Н. Цифровая обработка сигналов : методы и средства : учебн. пособие для вузов / В. Н. Бондарев, Г. Трестер, В. С. Чернега. – 2-е изд. – Х. : Конус, 2001. – 398 с.
5. Гонсалес Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. – К.: Техносфера, 2006. – 1073 с.
6. Gonzalez D. Digital Image Processing / D. Gonzalez, C. Rafael, C. Richard – 2nd ed. – published by Pearson Education, Inc. publishing as Prentice Hall, 2002. – 793 p.
7. Блаттер К. Вейвлет-анализ. Основы теории / К. Блаттер. – К.: Техносфера, 2006. – 279 с.
8. Кудряшов Б. Д. Конспект лекций по теории информации [Електронний ресурс] / Б. Д. Кудряшов. – Режим доступу: <http://k51.spb.ru/article.php?art=105&s=2>.
9. Gonzalez, R. and Woods, R.: Digital Image Processing. Published by Pearson, 1168 p. (2018).

10. Barannik, V., Krasnorutskiy, A., Ryabukha, Y., Okladnoy, D.: Model intelligent processing of aerial photographs with a dedicated key features interpretation. In: Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET), pp. 736-739. Lviv-Slavske (2016). doi: 10.1109/TCSET.2016.7452167.

11. Boyle, William S; Smith, George E.: Charge Coupled Semiconductor Devices. Bell Syst. Tech. J. 49 (4): 587–593. (1970)

12. Chigorin A., Krivovyaz G., Velizhev A., Konushin A.: A method for traffic sign detection in an image with learning from synthetic data. In: 14th International Conference Digital Signal Processing and its Applications, pp. 316-335. IEEE Press (2012)

13. Ding Z., Chen H., Gua Y., Peng Q.: GPU accelerated interactive space-time video matting. In: Computer Graphics International, pp. 163-168. (2010)

14. Козловський А. В. Комп'ютерна техніка та інформаційні технології: навч. посіб./ А. В. Козловський, Ю. М. Паночішин, В. В. Погрі-щук. – 2-ге видання., стер. – К.: Знання, 2012. – 463 с.

15. Конспект лекцій з дисципліни “Методи обробки мультимедійної інформації” для студентів усіх форм навчання за спец. 8.05090301 – “Інформаційні мережі зв'язку” / упоряд.: В. В. Бараннік, Н. А. Харченко. – Харків: ХНУРЕ, 2013. – 220 с.

16. Основи теорії та практики наукових досліджень: навч. посіб. / Ю. В. Стасєв, В. В. Бараннік, Ю. М. Рябуха, В. Є. Курілко. – Х.: ХНУПС, 2019. – 195 с.

17. Котомчак О. Ю. Комп'ютерна обробка зображень та мультимедіа : навч. посіб., метод. розроб. до лаб.робіт./

О. Ю. Котомчак – К. : Редакційно-видавничий центр Державного університету телекомунікацій, 2018. – 124 с.

18. Різуненко А. О. Теорія та практика цифрової обробки зображень: монографія. – Полтава: РВВ ПУСКУ, 2009. – 195 с.

19. Цифрова обробка аудіо- та відеоінформації у мультимедійних системах: навч. посіб. / О. В. Дробик, В. В. Кідалов, В. В. Коваль та ін. – К.: Наукова думка, 2008. – 144 с.

20. Цифрова обробка сигналів та зображень : навч. посіб. для студентів спеціальності 151 “Автоматизація та комп'ютерно-інтегровані технології” / уклад.: Тотосько О. В., Стухляк П. Д. – Тернопіль : Тернопільський національний технічний університет імені Івана Пулюя, 2016. – 140 с.

21. Gonzales R. C. Digital image processing / R. C. Gonzales, R. E. Woods. – Prentice Inc. Upper Saddle River, New Jersey 2002. – 779 p.

22. Lee S. Y. Temporally coherent video matting / S. Y. Lee, J. C. Yoon, I. K. Lee // Graphical Models 72. – 2010. – P. 25-33.

23. Loebl J. Image analysis: Principles and practice / J. Loebl. – London : Short Run press. – 1985. – 250 p.

### Розділ 3

1. Кодирование трехмерных моделей видеок кадров в инфотелекоммуникационных системах: монография / В. В. Бараннік, В. П. Поляков, А. В. Слободянюк под ред. В. В. Баранника. – Каменец-Подольский, 2011. – 210 с.

2. Баранник В. В. Кодирование трансформированных изображений в инфокоммуникационных системах / В. В. Баранник, В. П. Поляков. – Х. : ХУПС, 2010. – 212 с.

3. Barannik V., Babenko Y., Kulitsa O., Barannik V., Khimenko A. and Matviichuk-Yudina O. Significant Microsegment Transformants Encoding Method to Increase the Availability of Video Information Resource. *IEEE Advanced Trends in Information Theory (ATIT): proceedings of 2nd Intern. Conf.* (Kyiv, Ukraine, November 25-27, 2020). Kyiv, 2020. P. 52-56. DOI: 10.1109/ATIT50783.2020.9349256..
4. Баранник В. В. Структурно-комбинаторное представление данных в АСУ / В. В. Баранник, Ю. В. Стасев, Н. А. Королева. – Х.: ХУПС, 2009. – 252 с.
5. Миано Дж. Форматы и алгоритмы сжатия изображений в действии: пер. с англ.: учебн. пособие / Дж. Миано. – К. : Триумф, 2003. – 336 с.
6. Сэломон Д. Сжатие данных, изображений и звука / Д. Сэломон. – К.: Техносфера, 2004. – 368 с.
7. Golomb S. W. Run-length encodings / S. W. Golomb // *IEEE Transactions on Information Theory*. – 1966. – IT-12, № 3. – P. 399 – 401.
8. Elias P. Interval and recency rank source encoding: two on-line adaptive variable-rate schemes / P. Elias // *IEEE Transactions on Information Theory*. – 1987. – IT-33, № 1. – P. 3 – 10.
9. Шульгин В. И. Основы теории передачи информации. Ч. I. Экономное кодирование : учебн. пособие / В. И. Шульгин. – Х. : Нац. аэрокосм. ун-т “Харьк. авиац. ин-т”, 2003. – 102 с.
10. Rissanen J. J. Universal coding, information, prediction, and estimation / J. J. Rissanen // *IEEE Transactions on Information Theory*. – 1984. – Vol. 30, № 4. – P. 629 – 636.
11. Welch T. A. A technique for high-performance data compression / T. A. Welch // *IEEE Computer*. – June, 1984. – Vol. 17, № 6. – P. 8 – 19.

12. Ziv J. A universal algorithm for sequential data compression / J. Ziv, A. Lempel // *IEEE Transactions on Information Theory*. – 1977. – Vol. 23, № 3. – P. 337 – 343.
13. Ziv J. Compression of individual sequences via variable-rate coding / J. Ziv, A. Lempel // *IEEE Transactions on Information Theory*. – 1978. – Vol. 24, № 5. – P. 530 – 536.
14. Alay O. and Stephansen H.: The effect of multi-generation encoding in broadcast contribution on the end-user video quality. In: *Proc. 19th Int. Packet Video Workshop (PV)*, pp. 113-118, (2012).
15. Ahmad I., Wei X., Sun Y. and Zhang Y.-Q.: Video transcoding: An overview of various techniques and research issue. In: *IEEE Trans. Multimedia*, vol. 7, no. 5, pp. 793-804, (2005).
16. Baccouch H., Ageneau P. L., Tizon N., Boukhatem N.: Prioritized network coding scheme for multilayer video streaming. In: *14th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pp. 802-809, IEEE Press (2017)
17. Bai X., Wang J.: Towards temporally-coherent video matting. In: *Proceedings of the 5th international conference on Computer vision/computer graphics collaboration techniques*, pp. 63-74. MIRAGE'11, Springer-Verlag (2011)
18. Barannik V., Podlesny S., Krasnorutskyi A., Musienko A., Himenko V.: The ensuring the integrity of information streams under the cyberattacks action. In: *East-West Design & Test Symposium (EWDTS)*, pp. 1-5. IEEE Press (2016) doi: 10.1109/EWDTS.2016.7807752.
19. Barannik V., Podlesny S., Tarasenko D., Barannik D., Kulitsa O.: The video stream encoding method in infocommunication systems. In: *Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET)*, pp. 538-541. Lviv 2018. doi: 10.1109/TCSET.2018.8336259.
20. Barannik V.V., Ryabukha Yu. N., Tverdokhlebov V.V., Barannik D.V.: Methodological basis for constructing a method for

compressing of transformants bit representation, based on non-equilibrium positional encoding. In: *Advanced Information and Communication Technologies (AICT)*, 2017 2nd International Conference, pp.188-192. (2017). doi: 10.1109 / AIACT.2017.8020096.

21. Бараннік В. В. Метод кодування бітового опису трансформант на базі незалежних кодових структур у технології управління інтенсивністю відеопотоку / В. В. Бараннік, Ю. М. Рябуха, В. В. Твердохліб та ін. // *Наукоємні технології*. – К., 2019. – № 2(42). – С. 295 – 299.

22. Barannik V. Complex Method of Video Stream Intensity Control. *Engineer of the XXI Century* / V. Barannik, V. Himenko, V. Tverdokhleb, S. Kovalevskiy. – 2020. – P. 209-215.

23. Birchfield S. An Introduction to Projective Geometry (for computer vision) / S. Birchfield. – URL: <http://www.dai.ed/Cvonline/geom.htm>.

24. Barannik V., Barannik D., Fustii V., Parkhomenko M. Evaluation of Effectiveness of Masking Methods of Aerial Photographs. *Advanced Information and Communications Technologies (AICT)*: proceedings of 3rd Intern. Conf. (Lviv, Ukraine, July 2-6, 2019), Lviv, 2019. DOI: 10.1109/AIACT.2019.8847820.

25. Barannik V., Barannik V., Havrylov D., Sorokun A. Development Second and Third Phase of the Selective Frame Processing Method. *Advanced Information and Communications Technologies (AICT)*: proceedings of 3rd Intern. Conf. (Lviv, Ukraine, July 2-6, 2019), Lviv, 2019. P. 54-57. DOI: 10.1109/AIACT.2019.8847897.

26. Barannik V., Shulgin S. The method of increasing accessibility of the dynamic video information resource. *Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET)*: proceedings of 13th Intern. Conf. (Lviv,

Ukraine, February 23-26, 2016), Lviv, 2016. P. 621-623. DOI: 10.1109/TCSET.2016.7452133.

27. Barannik V., Tarasenko D. Method coding efficiency segments for information technology processing video. *Problems of Infocommunications. Science and Technology (PIC S&T)*: proceedings of 4th International Scientific-Practical Conference. (Kharkov, Ukraine, October 10-13, 2017), Kharkov, 2017. P. 551-555. DOI: 10.1109/INFOCOMMST.2017.8246460.

28. Barannik V., Tverdokhlib V., Ryabukha Yu., Dodukh A., Suprun O., Tarasenko D. Integration the non-equilibrium position encoding into the compression technology of the transformed images. *IEEE East-West Design & Test Symposium: 15th IEEE Intern. Symp.*, (Novi Sad, Serbia, September 29–October 2, 2017). Novi Sad: 2017. P. 337-339.

29. Barannik V.V., Tverdokhleb V.V., Ryabukha Yu. N., Barannik D.V. Methodological basis for constructing a method for compressing of transformants bit representation, based on non-equilibrium positional encoding. *Advanced Information and Communication Technologies (AICT)*, 2017 2nd International Conference. pp.188-192. (2017). DOI: 10.1109/AIACT.2017.8020096.

30. Barannik V.V., Tverdokhlib V.V., Ryabukha U.N., Shulgin S.S., Barannik D.V. Video segment coding method for bit rate control information technology. *Science-Based Technologies*. 2020. Vol. 46. Issue 3. P. 316-321.

31. Barannik V., Karpenko S. Method of the 3-D image processing. *Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET)*: proceedings of Intern. Conf. (Lviv-Slavsko, Ukraine, February 19-23, 2008), Lviv-Slavsko, 2008. P. 378-380.

32. Barannik V., Hahanova A., Krivonos V. Coding tangible component of transforms to provide accessibility and integrity of video data. *East-West Design & Test Symposium (EWDTS)*: proceedings of Intern. Conf. (Rostov-on-Don, Russia, September 27-30, 2013), Rostov-on-Don, 2013. DOI: 10.1109/EWDTS.2013.6673179.

33. Barannik N., Tverdokhlib V., Evgeniy E., Zakomorna K., Vorontsova N. and Shmakov V. Constructing the Code Description Of the Transformant DCT Taking into Account their Structural Features. *IEEE Advanced Trends in Information Theory (ATIT)*: proceedings of 2nd Intern. Conf. (Kyiv, Ukraine, November 25-27, 2020). Kyiv, 2020. P. 34-37. DOI: 10.1109/ATIT50783.2020.9349255.

34. Barannik V., Belikova T., Gurzhii P. The model of threats to information and psychological security, taking into account the hidden information destructive impact on the subconscious of adolescents. *IEEE Advanced Trends in Information Theory (ATIT)*: proceedings of Intern. Conf. (Kyiv, Ukraine, December 18-20, 2019). Kyiv, 2019. P. 656-661. DOI: 10.1109/ATIT49449.2019.9030432.

35. Barannik V., Karpinski M., Tverdokhle V., Barannik D., Himenko V, Marek Aleksander. The technology of the video stream intensity controlling based on the bit-planes recombination. *IEEE Wireless Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS)*: proceedings of 4th International Symposium. (Lviv, Ukraine, September 20-21, 2018), Lviv, 2018. DOI: 10.1109/IDAACS-SWS.2018.8525560.

36. Barannik V., Shulgin S., Krasnorutsky A., Slobodanyuk O., Gurzhii P. and Korolyova N. Methodological Funda-

mentals of Deciphering Coding of Aerophotography Segments on Special Equipment of Unmanned Complex. *IEEE Advanced Trends in Information Theory (ATIT)*: proceedings of 2nd Intern. Conf. (Kyiv, Ukraine, November 25-27, 2020). Kyiv, 2020. P. 38-43. DOI: 10.1109/ATIT50783.2020.9349257.

37. Barannik V., Sidchenko S. and Barannik D. Technology for Protecting Video Information Resources in the Information Communication Space. *IEEE Advanced Trends in Information Theory (ATIT)*: proceedings of 2nd Intern. Conf. (Kyiv, Ukraine, November 25-27, 2020). Kyiv, 2020. P. 29-33. DOI: 10.1109/ATIT50783.2020.9349324.

38. Barannik V., Tupitsya I., Stepanko O., Kovalenko O., Yroshenko V. and Sidchenko Y. The Analysis of the Internal Restructuring Method Efficiency Used For a More Compact Representation of the Encoded Data. *IEEE Advanced Trends in Information Theory (ATIT)*: proceedings of 2nd Intern. Conf. (Kyiv, Ukraine, November 25-27, 2020). Kyiv, 2020. P. 48-51. DOI: 10.1109/ATIT50783.2020.9349307.

39. Barannik, V., Krasnoruckiy, A. and Hahanova, A. The positional structural-weight coding of the binary view of transformants. *East-West Design & Test Symposium (EWDTS)*, pp. 1-4. Rostov-on-Don (2013) DOI: 10.1109/EWDTS.2013.6673178.

40. Barannik, V.V., Ryabukha, Yu.N. and Kulitsa, O.S. The method for improving security of the remote video information resource on the basis of intellectual processing of video frames in the telecommunication systems. *Telecommunications and Radio Engineering*. Vol. 76. No 9. pp. 785-797. (2017). DOI: 10.1615/TelecomRadEng.v76.i9.40

41. Barannik, V. and Barannik, N. and Ryabukha, Yu. and Barannik, D. Indirect Steganographic Embedding Method Based

On Modifications of The Basis of the Polyadic System. 15th IEEE International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET'2020), pp. 699-702 (2020) DOI: 10.1109/TCSET49122.2020.235522.

42. Barannik, V. and Barannik, V. Binomial-Polyadic Binary Data Encoding by Quantity of Series of Ones. 15th IEEE International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET'2020), pp. 775-780 (2020) DOI: 10.1109/TCSET49122.2020.235540.

43. Barannik, D. Stegano-Compression Coding in a Non-Equalible Positional Base. *IEEE Advanced Trends in Information Theory (ATIT)*: proceedings of 2nd Intern. Conf. (Kyiv, Ukraine, November 25-27, 2020). Kyiv, 2020. P. 83-86. DOI: 10.1109/ATIT50783.2020.9349328.

**Д Л Я   Н О Т А Т О К**

*Навчальне видання*

**Бараннік** Володимир Вікторович

**Рябуха** Юрій Миколайович

**Твердохліб** Віталій Вікторович

**ОСНОВИ ОБРОБКИ ТА ЗАХИСТУ ВІДЕОІНФОРМАЦІЇ  
В ІНФОРМАЦІЙНИХ СИСТЕМАХ**

Навчальний посібник

для курсантів та студентів вищих навчальних закладів

Дизайн обкладинки *В. В. Бараннік, В.В. Твердохліб*

Формат 60x84/16. Ум. друк. арк. 10,00. Тир. 500 прим. Зам. 831-21.

Видавець та виготовлювач ФОП Бровін О.В.

61022, м. Харків, вул. Трінклера, 2, корп.1, к.19. Т. (066) 822-71-30

Свідоцтво про внесення суб'єкта до Державного реєстру  
видавців та виготовників видавничої продукції серія ДК 3587 від 23.09.09 р.

---

---

**СТИЛЬ**®  
**ИЗДАТ**  
ТИПОГРАФІЯ  
www.stil-izdat.com