

# **КОМП'ЮТЕРНА ОБРОБКА ЗОБРАЖЕНЬ**

**Digital Image Processing - DIP**

**2019 / 2020 навчальний рік**

# **МОДУЛЬ 3. Стиснення зображень**

**3.1. Загальні відомості зберігання зображень. Урахування особливостей зору.**

**3.2. Алгоритми стиснення без витрат.**

**3.3. Стиснення з витратами.**

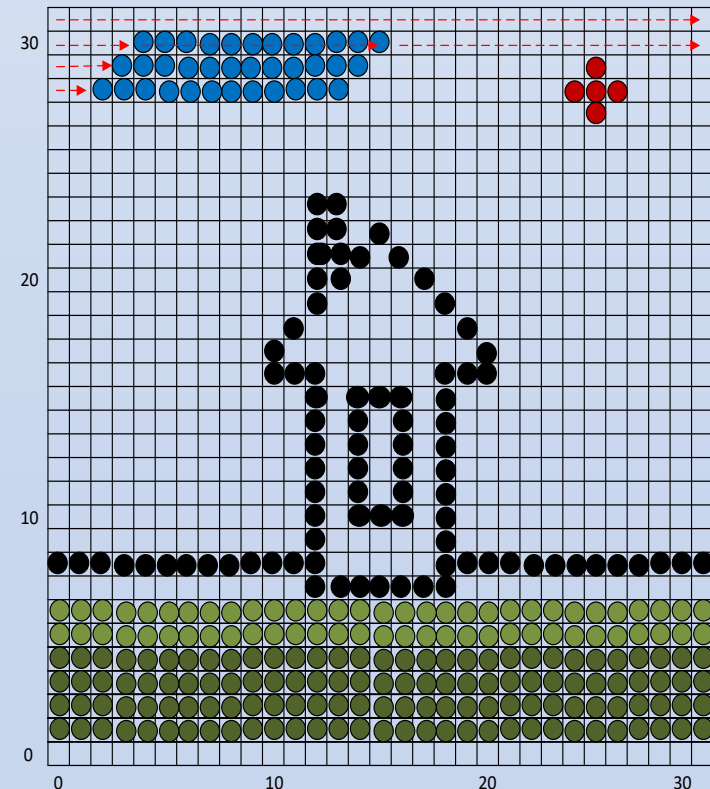
# Алгоритм групового кодування (RLE - Run Length Encoding)

Ланцюжок однакових байтів



Лічильник повторень : RGB

Перший варіант  
алгоритму - РСХ



# Ознака лічильника - одиниці в двох верхніх бітах зчитаного байта:



# Другий варіант алгоритму (TIFF, TGA)

Лічильник повторень : RGB

Лічильник (8 біт)

0 XXXXXXXX

Пропуск

Лічильник (8 біт)

1 XXXXXXXX

RGB

# Характеристики алгоритму RLE:

Коефіцієнти компресії:

	Кращий	середній	гірший
1 варіант:	32,	2,	0,5.
2 варіант:	64,	3,	128/129.

Клас зображень: Зображення з невеликою кількістю кольорів (ділова і наукова графіка).

Симетричність: Близько 1.

# Алгоритм LZW (розробники Lempel, Ziv і Welch).

Стиснення здійснюється за рахунок однакових ланцюжків байт.

LZW універсальний - саме його варіанти використовуються в звичайних архіваторах.

LZW реалізований в форматах GIF і TIFF.

# Характеристики алгоритму LZW:

Коефіцієнти компресії:

Кращий	середній	гірший
--------	----------	--------

1000	57	4
------	----	---

Симетричність: Майже симетричний, за умови оптимальної реалізації



# Алгоритм Хаффмана

Класичний алгоритм, відомий з 60-х років ХХ століття.

Зіставляє символам вхідного потоку, які зустрічаються більшу кількість разів, ланцюжок біт меншої довжини, і навпаки.

Для збору статистики вимагає двох проходів по зображенню.

# Алгоритм Хаффмана

**Визначення.** Нехай заданий алфавіт  $A$

$$A = \{\alpha_1, \alpha_2, \dots, \alpha_n\}$$

що складається з кінцевого числа  $n$  різних символів. Алфавіту відповідає множина ваг  $W$

$$W = \{w_1, w_2, \dots, w_n\}$$

цілих позитивних чисел – визначених як кількість  $w_i$  символів  $\alpha_i$  в деякому тексті (зображенні).

**Код Хаффмана**  $C = \{c_1, c_2, \dots, c_n\}$ , де  $c_i$  – код символу  $\alpha_i$ , є такий що:

- $c_i$  – не є префіксом для  $c_j$  при  $c_i \neq c_j$ ,
- сума  $\sum_{i \in [1, n]} w_i * |c_i|$  мінімальна, де  $|c_i|$  – довжина  $L$  коду  $c_i$ .

# Алгоритм побудови кодуєчої схеми

**Крок 0.** Розрахунок ваг (ймовірностей)  $w_i$  появи букв вхідного алфавіту  $\alpha_i$  у вхідному потоці.

**Крок 1.** Впорядкування всіх букв вхідного алфавіту в порядку убутання ваг (ймовірностей)  $w_i$ .

# Алгоритм побудови кодуєчої схеми

**Крок 2.** Об'єднуються два символи  $\alpha_{(r-1)}$  і  $\alpha_{(r)}$  з найменшими вагами  $w_{(r-1)}$  і  $w_{(r)}$  в псевдосимвол  $\alpha^r\{\alpha_{(r-1)}, \alpha_{(r-1)}\}$  з вагою  $(w_{(r-1)} + w_{(r)})$   
!! Допикується:

0 в початок коду  $\mathbf{c}_{(r-1)} = "0" + \mathbf{c}_{(r-1)}$

1 в початок коду  $\mathbf{c}_{(r)} = "1" + \mathbf{c}_{(r)}$

# Алгоритм побудови кодуєчої схеми

Крок 3. Видалення зі списку  
впорядкованих символів  $\alpha_{(r-1)}$  і  $\alpha_{(r)}$ ,  
та занесення туди псевдосимвола  
 $\alpha^r\{\alpha_{(r-1)}, \alpha_{(r-1)}\}$

Кроки 2, 3 повторюються до тих пір,  
поки в списку не залишиться один  
псевдосимвол (з вагою = 1).

# Алгоритм побудови кодуєчої схеми (приклад)

Нехай є 6 букв в алфавіті:

з вагами :

$$a_1 \rightarrow w_1 = 0,4 ; a_2 \rightarrow w_2 = 0,25;$$

$$a_3 \rightarrow w_3 = 0,15 ; a_4 \rightarrow w_4 = 0,1;$$

$$a_5 \rightarrow w_5 = 0,06 ; a_6 \rightarrow w_6 = 0,04;$$

Тоді процес побудови схеми можна уявити  
так:

# Алгоритм побудови кодуєчої схеми (приклад)

Сортуємо символи на кожному кроці

Символи	Ваги символів					
$a_1$	0,40	0,40	0,40	0,40	0,60	1,00
$a_2$	0,25	0,25	0,25	0,35	0,45	
$a_3$	0,15	0,15	0,20	0,20		
$a_4$	0,10	0,10	0,15			
$a_5$	0,06	0,10				
$a_6$	0,04					

Будуємо кодове бінарне дерево

# Алгоритм побудови кодуєчої схеми (приклад)

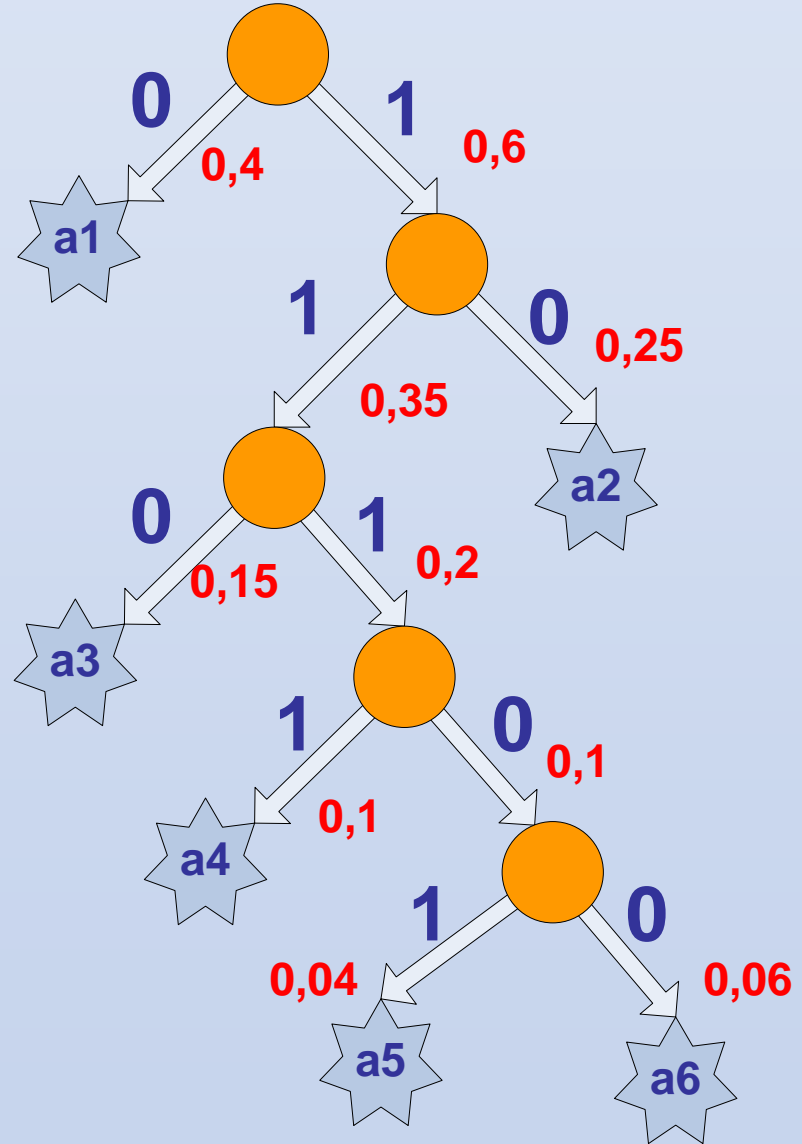
Для побудови кодуєчих слів треба пройти по стрілках від початкових символів до кінця отриманого бінарного дерева:

a1 = 0, a2 = 10

$$a_3 = 110, a_4 = 1111$$

a5 – 11101

a6 – 11100





# Алгоритм Хаффмана

При стисненні зображень в якості букв вхідного алфавіту використовуються кольори нестиснутого зображення

**Приклад.** Зображення 100x100, 8біт / піксель

Загальний обсяг 80000 біт колірної інформації.

У зображенні:

**5000** синіх точок ( $w_1=0.5$ )

**2400** червоних точок ( $w_2=0.24$ )

**1500** жовтих точок ( $w_3=0.15$ )

**1100** зелених точок ( $w_4=0.11$ )

# Алгоритм Хаффмана

## Кодування:

Символи	Ваги символів				
$col_1$	0,5	0,50	0,5 <b>0</b>	Синій	0
$col_2$	0,24	0,26 <b>0</b>	0,5 <b>1</b>	Червоний	11
$col_3$	0,15 <b>0</b>	0,24 <b>1</b>		Жовтий	100
$col_4$	0,11 <b>1</b>			Зелений	101

Обсяг колірної інформації:

$$5000 + 2400 * 2 + (1500 + 1100) * 3 = 17\,600 \text{ біт}$$

замість 80 000

# Характеристики класичного алгоритму Хаффмана:

Коефіцієнти компресії:

кращий - 8,

середній - 1,5,

гірший - 1

Симетричність: 2

# Алгоритм Хаффмана

Алгоритм Хаффмана - єдиний з усіх не збільшує розміру вихідних даних в гіршому випадку (якщо не брати до уваги необхідність зберігати таблицю перекодування разом з файлом).

Практично не застосовується до зображень в чистому вигляді.

Зазвичай використовується як один з етапів компресії в більш складних схемах.

# Алгоритм CCITT Group 3

При стисненні чорно-білих зображень (один біт на піксель) використовується модифікація алгоритму, яка називається CCITT Group 3 (Consultative Committee International Telegraph and Telephone - третя група по стандартизації Міжнародного Консультаційного Комітету з телеграфного й телефонного зв'язку).

Коефіцієнти компресії: в кращому випадку - до 213, в середньому - 2, в гіршому випадку збільшує файл в 5 разів.

Симетричність: Близька до 1.

Клас зображень: чорно-білі зображення, із значно переважним білим (тексти).

# Рекомендована ЛІТЕРАТУРА

- **Вовк С.М., Гнатушенко В.В., Бондаренко М.В.** Методи обробки зображень та комп'ютерний зір: навчальний посібник. - Д.: Ліра, 2016 — 148 с.
- **Красильников Н.Н.** Цифровая обработка 2D- и 3D-изображений: учеб.пособие.- СПб.: БХВ-Петербург, 2011.- 608 с.: ил.
- **Гонсалес Р.С., Вудс Р.Э.** Цифровая обработка изображений. - М. : Техносфера, 2005. -1070 с.
- **Визильтер Ю.В., Желтов С.Ю. и др.** Обработка и анализ изображений в задачах машинного зрения.-М.: Физматкнига, 2010.-672 с.

## Рекомендована ЛІТЕРАТУРА

- **Ватолин Д., Ратушняк А., Смирнов М., Юкин В.** Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео. - М.: ДИАЛОГ-МИФИ, 2002. - 384 с.
- **Творошенко І.С.** Конспект лекцій з дисципліни «Цифрова обробка зображень» / І.С.Творошенко : І. С. Творошенко ; Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова. – Харків : ХНУМГ ім. О. М. Бекетова, 2017. – 75 с.
- Методы компьютерной обработки изображений: Учебное пособие для ВУЗов/ Под ред.: **Сойфер В.А..** - 2-е изд., испр. - М.: Физматлит, 2003. - 780 с.
- **Фисенко В.Т., Фисенко Т.Ю.** Компьютерная обработка и распознавание изображений: учеб. пособие. - СПб: СПбГУ ИТМО, 2008. – 192 с.

## Додаткова ЛІТЕРАТУРА

- **Грузман И.С., Киричук В.С.** Цифровая обработка изображений в информационных системах. — Новосибирск: Изд-во НГУ, 2002. — 352 с.: ил.
- **Solomon C., Breckon T.** Fundamentals of Digital Image Processing. — Willey-Blackwell, 2011 - 344 p.
- **Павлидис Т.** Алгоритмы машинной графики и обработки изображений: Пер. с англ. - М.: Радио и связь, 1986. — 400 с.
- **Яншин В. В., Калинин Г. А.** Обработка изображений на языке Си для IBM PC: Алгоритмы и программы. — М.: Мир, 1994. — 240 с.



# Інформаційні ресурси

- Компьютерная обработка изображений. Конспект лекций. [http://aco.ifmo.ru/el\\_books/image\\_processing/](http://aco.ifmo.ru/el_books/image_processing/)
- Цифрова обробка зображень [Електронний ресурс] : методичні рекомендації до виконання лабораторних робіт / НТУУ «КПІ» ; уклад.: В. С. Лазебний, П. В. Попович. – Електронні текстові дані (1 файл: 1,41 Мбайт). – Київ : НТУУ «КПІ», 2016. – 73 с. – <https://ela.kpi.ua/handle/123456789/21035>
- <https://www.youtube.com/watch?v=CZ99Q0DQq3Y>
- <https://www.youtube.com/watch?v=FKTLW8GAdu4>

# **The END**

## **Modulo 3.2**