УДК 681.3.04

І.А. Дичка, В.І. Голуб, М.В. Новосад

МЕТОД УЩІЛЬНЕННЯ АЛФАВІТНО-ЦИФРОВОЇ ІНФОРМАЦІЇ, ПОДАНОЇ В ГРАФІЧНОКОДОВАНОМУ ВИГЛЯДІ

We devise the compression method of alphanumeric information in its graphic coding representation. The method of data compression is based on converting data from one alphabet to another – from the alphabet of characters into alphabet of graphic coding symbols. The method provides the data compression on averageby 20% for digit data and by 12 % for text sequences. Furthermore, this method provides the storage of a largeinformation content (over 1000 alphanumerical characters) in the form of graphical codes on a limited area of carrier.

Вступ

Подання інформації у графічнокодованому вигляді є одним із напрямів підвищення швидкості та надійності введення даних в обчислювальну систему [1, 2]. Мета графічного кодування – забезпечити автоматичне введення досить великих обсягів даних, використовуючи нескладні та недорогі технічні засоби (сканери) [3, 4].

Графічнокодовані дані подаються у вигляді графічного коду (ГК). ГК – це двовимірний масив дискретних графічних елементів, оформлених як єдине ціле (графічний файл). Графічними елементами можуть бути: квадрат, трикутник, многокутник, штрих, круг, еліпс тощо [1]. Конструктивне оформлення графічного коду як двовимірного масиву дискретних елементів називають графічнокодовою позначкою (ГК-позначкою) [5, 6].

Зчитування ГК-позначок з носія здійснюється оптичним способом, у тому числі й на відстані.

Завдяки властивості автоматичного зчитування ГК набули поширення в різних галузях людської діяльності: промисловому виробництві, системах документообігу, транспортній сфері, поштовій галузі тощо.

ГК розміщують на поверхні об'єкта обліку, він є ідентифікатором об'єкта і переміщується разом із ним по всій траєкторії його руху.

У сучасних системах автоматичної ідентифікації на основі графічного кодування даних вимагається, щоб ГК подавав від кількох сотень до кількох тисяч алфавітно-цифрових символів, тобто був не лише ідентифікатором об'єкта, а й слугував би своєрідною переносною базою даних (portable data file). Досягти цього можна лише забезпеченням високої інформаційної щільності даних, що підлягають поданню у графічнокодованому вигляді.

Існує кілька стандартів двовимірних графічних кодів – Data Matrix, MaxiCode, Array Tag, Aztec Code, QR Code, у яких тою чи іншою мірою використовуються певні засоби ущільнення даних [4–6]. Їх характерною особливістю є те, що ущільнення досягається лише при роботі з латиномовними текстами. Якщо зазначені стандарти кодування застосовуються для подання у графічнокодованому вигляді інформації на основі нелатинського алфавіту, зокрема кирилиці, то досягти ущільнення даних неможливо.

Постановка задачі

Дослідження спрямоване на розроблення узагальненого методу, який би при графічнокодованому поданні забезпечував ефективне ущільнення даних, отриманих з використанням довільного алфавіту.

Визначення потужності алфавіту для ущільненого подання алфавітно-цифрових даних

Розглядатимемо випадок чорно-білих двовимірних графічних кодів (рис. 1).

При створенні ГК-позначок двовимірних графічних кодів доцільно використовувати бітове подання даних – коли результуючу послідовність даних подають у двійковому вигляді і двійковому нулю ставлять у відповідність світ-

Рис. 1. Приклад ГК-позначки двовимірного графічного коду

лий елемент, а одиниці – темний елемент графічнокодового зображення.

У графічнокодованому вигляді можуть подаватися будь-які алфавітно-цифрові дані, що належать розширеному ASCII (комп'ютерному алфавіту) [1–3]. Однак при створенні певної множини алфавітно-цифрових повідомлень може використовуватись обмежений набір символів з розширеного ASCII, який назвемо алфавітом повідомлень.

Якщо потужність алфавіту повідомлень менша за 256 – потужність розширеного ASCII, і відмінна від степеня двійки, то з'являється можливість закодувати повідомлення з ущільненням даних.

Ущільненням даних називатимемо таке кодування алфавітно-цифрових повідомлень (тексту), при якому для повідомлення завдовжки h алфавітно-цифрових символів, що використовує алфавіт А потужністю P_{A}, довжина $B(h)$ результуючої двійкової послідовності, отриманої внаслідок певного перетворення вихідного тексту, задовольняє умову

$$B(h) < h]\log_2 P_{\text{A}}[. \qquad (1)$$

Для неущільнених повідомлень маємо

$$B(h) = h]\log_2 P_{\text{A}}[.$$

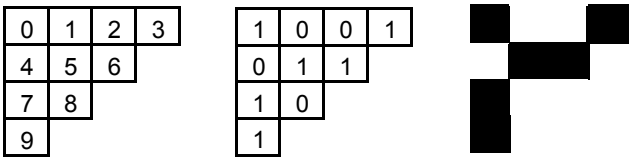
Ступінь ущільнення алфавітно-цифрових даних оцінюватимемо коефіцієнтом ущільнення

$$U(P_{\text{A}}) = \frac{h]\log_2 P_{\text{A}}[}{B(h)}.$$

Завдання полягає в тому, щоб знайти такий спосіб перетворення вхідного потоку алфавітно-цифрових даних на двійкову послідовність $B(h)$ та визначити таку потужність P_{A} алфавіту повідомлень, при яких коефіцієнт ущільнення $U(P_{\text{A}})$ був би якомога більшим ($U(P_{\text{A}}) > 1$).

Для того щоб об'єднати можливість використання бітового способу заповнення матриці ГК-позначки графічного коду і потребу забезпечення завадостійкості даних на основі коректувального коду Ріда–Соломона, використовують поле Галуа з розмірністю $q = 2^s$. При цьому операції над елементами поля GF(2^s) виконують за модулем незвідного многочлена степеня s.

При формуванні бітової матриці ГК-позначки зручно користуватись ГК-знаками (рис. 2). ГК-знак є мінімальною структурною одиницею ГК-позначки (ГК-позначка складається з ГК-знаків). Доцільно, щоб ГК-знак складався з s комірок і, отже, подавав один q-значний ($q = 2^s$) розряд кодового слова коду Ріда–Соломона. Тоді множина всіх можливих ГК-знаків утворить алфавіт Ω графічного коду потужністю $P_\Omega = 2^s$, який називають символікою графічного коду. s-розрядний двійковий код, що відповідає ГК-знаку, називатимемо кодовектором.



Порядок розміщення бітів у ГК-знаку (9 – старший розряд, 0 – молодший розряд)

Бітова карта ГК-знака, що відповідає кодовектору 1011101001

Зображення кодовектора на носії

Рис. 2. Приклад ГК-знака при $s = 10$

ГК-знаки наносяться на носій інформації у вигляді матриці.

Алфавіт Ω складається з інформаційних $\Omega_{\text{інф}}$ та службових $\Omega_{\text{сл}}$ ГК-знаків: $\Omega = \{\Omega_{\text{інф}}\} \cup \cup \{\Omega_{\text{сл}}\}$. Інформаційні ГК-знаки (їм відповідають інформаційні кодовектори) $\Omega_{\text{інф}}$ використовуються для подання на носії вхідних алфавітно-цифрових послідовностей, службові ГК-знаки (їм відповідають службові кодовектори) $\Omega_{\text{сл}}$ – для настроювання сканера, позиціювання ГК-позначки на носії, перемикання режимів кодування даних тощо.

Нехай $P_{\Omega_{\text{інф}}}$ – кількість інформаційних, а $P_{\Omega_{\text{сл}}}$ – кількість службових кодовекторів, тоді $P_{\Omega_{\text{інф}}} = P_\Omega - P_{\Omega_{\text{сл}}}$. Процес ущільнення передбачає роботу лише з інформаційними кодовекторами.

Таким чином, кодування алфавітно-цифрової послідовності, що використовує алфавіт А потужністю P_{A}, зводиться до перетворення символів алфавіту А на символи алфавіту $\Omega_{\text{інф}}$ потужністю $P_{\Omega_{\text{інф}}}$.

У загальному випадку таке перетворення має вигляд

$$n(P_{\text{A}}) \to m(P_{\Omega_{\text{інф}}}), \qquad (2)$$

тобто n символів алфавіту А перетворюються на m символів алфавіту $\Omega_{\text{інф}}$.

Вираз типу (2) реалізуємо як перетворення числа з однієї системи числення в іншу, а саме перетворення n-розрядного числа з системи числення з основою P_{A} на m-розрядне число системи числення з основою $P_{\Omega_{\text{інф}}}$:

$$\sum_{i=0}^{n-1} a_i P_{\text{A}}^i \to \sum_{j=0}^{m-1} \omega_j P_{\Omega_{\text{інф}}}^j, \qquad (3)$$

де a_i – числове значення символу з алфавіту А (його порядковий номер в алфавіті А), ω_j – числове значення символу з алфавіту $\Omega_{\text{інф}}$, $\omega_j \in \{0, 1, 2, ..., P_{\Omega_{\text{інф}}} - 1\}$.

Із врахуванням того, що максимальне значення n-розрядного числа в системі числення з основою P_{A} дорівнює $P_{\text{A}}^n - 1$, а максимальне значення m-розрядного числа в системі числення з основою $P_{\Omega_{\text{інф}}}$ дорівнює $P_{\Omega_{\text{інф}}}^m - 1$, перетворення (3) можливе, якщо

$$P_{\text{A}}^n - 1 \le P_{\Omega_{\text{інф}}}^m - 1$$

або

$$P_{\text{A}}^n \le P_{\Omega_{\text{інф}}}^m. \qquad (4)$$

Довжина n-розрядного числа в системі числення з основою P_{A} становить $n]\log_2 P_{\text{A}}[$ двійкових розрядів, а довжина m-розрядного числа в системі числення з основою $P_{\Omega_{\text{інф}}}$ – $m]\log_2 P_{\Omega_{\text{інф}}}[$ двійкових розрядів.

Таким чином, для того щоб перетворення (3) забезпечувало ущільнення алфавітно-цифрових даних, необхідно і достатньо, щоб виконувалась умова

$$\begin{cases} P_{\text{A}}^n \le P_{\Omega_{\text{інф}}}^m, \\ n]\log_2 P_{\text{A}}[> m]\log_2 P_{\Omega_{\text{інф}}}[, \end{cases} \qquad (5)$$

де $m]\log_2 P_{\Omega_{\text{інф}}}[$ – довжина ущільненої (результуючої) послідовності, яка побітово буде подана в графічнокодованому вигляді, а $n]\log_2 P_{\text{A}}[$ – довжина неущільненого повідомлення.

Виконання умови (5) при фіксованому (вибраному) $P_{\Omega_{\text{інф}}}$ зводиться до знаходження P_{A} – потужності алфавіту А, тобто до знаходження основи P_{A} відповідної системи числення, при якій буде забезпечуватися ущільнення.

Тоді коефіцієнт ущільнення матиме вигляд

$$U_s^{(P_{\Omega_{\text{інф}}})}(P_{\text{A}}) = \frac{n]\log_2 P_{\text{A}}[}{m]\log_2 P_{\Omega_{\text{інф}}}[}. \qquad (6)$$

Він показує, у скільки разів довжина двійкового рядка, що відповідає алфавітно-цифровій послідовності завдовжки n символів з алфавіту А ($n]\log_2 P_{\text{A}}[$), більша за довжину двійкового рядка (за $m]\log_2 P_{\Omega_{\text{інф}}}[$ двійкових розрядів), отриманого після виконання перетворення виду $n(P_{\text{A}}) \to m(P_{\Omega_{\text{інф}}})$.

Нехай s = 10, тобто P_Ω = 1024, і використовується поле Галуа GF(2^{10}), елементами якого є числа від 0 до 1023. Операції над елементами такого поля виконуватимемо за модулем незвідного багаточлена $m_{10}(x)$ степеня 10, наприклад:

$$m_{10}(x) = x^{10} + x^9 + x^8 + x^6 + x^4 + x^2 + 1.$$

Загалом існує 55 незвідних багаточленів степеня 10 і будь-який з них можна вибрати для побудови поля GF(2^{10}) [2].

Зарезервуємо $P_{\text{сл}}$ = 11 службових символів. Тоді кількість інформаційних слів $P_{\Omega_{\text{інф}}} = P_\Omega - 11 = 1013$.

Для $P_{\Omega_{\text{інф}}}$ = 1013 система (5) зводиться до вигляду

$$\begin{cases} P_{\text{A}}^n \le 1013^m, \\ n]\log_2 P_{\text{A}}[> 10m. \end{cases} \qquad (7)$$

Розв'яжемо систему (7) при невеликих m, а саме при m = 1, 2, 3, 4, 5, що випливає з практичних потреб кодування даних (табл. 1). Найбільш суттєві значення коефіцієнта ущільнення досягаються при семи значеннях потужності алфавіту P_{A} = 3, 5, 10, 17, 43, 75, 140 (див. рис. 3). Для використання алфавітів з потужностями P_{A} = 3, 5 вхідна алфавітно-цифрова послідовність має набувати трійкової або п'ятіркової форми, що неприйнятно. На практиці доцільно використовувати алфавіти: десятковий для подання цифрових послідовностей; P_{A} = 43, 75, 140 – для подання текстових даних. Оскільки найбільше значення коефіцієнта ущільнення $U(P_{\text{A}})$ = 1,12 досягається при P_{A} = 75, 140, то будь-яке з цих значень можна

***Таблиця* 1.** Найбільш доцільні для практичного застосування значення потужностей алфавіту P_A, за яких досягається ущільнення даних при $s = 10$ ($P_\Omega = 1024$, $P_{\Omega\ інф} = 1013$ і $P_{сл} = 11$)

Потужність алфавіту P_A	Коефіцієнт ущільнення $U_{10}^{(1013)}$	Тип перетворення "n" → "m"	Коментар
10	1,20	"3" → "1"	Три суміжні цифрові символи перетворюються на 1 кодовектор з поля GF(1024)
43	1,10	"11"→ "6"	11 суміжних символів з алфавіту потужністю $P_A = 43$ перетворюються на 6 кодовекторів з поля GF(1024)
75	1,12	"8" → "5"	8 суміжних символів з алфавіту потужністю $P_A = 75$ перетворюються на 5 кодовекторів з поля GF(1024)
140	1,12	"7" → "5"	7 суміжних символів з алфавіту потужністю $P_A = 140$ перетворюються на 5 кодовекторів з поля GF(1024)

Рис. 3. Залежність коефіцієнта ущільнення від потужності алфавіту P_A при $s = 10$ ($P_\Omega = 1024$, $P_{сл} = 11$)

вибрати як потужність алфавіту. Але з метою включення якомого більшої кількості символів в алфавіт текстового режиму покладаємо $P_A = 140$.

Розв'язавши систему (5) при $P_{\Omega\ інф} = 256$ ($s = 8$), 512 ($s = 9$), 1024 ($s = 10$), 2048 ($s = 11$), 4096 ($s = 12$), можна зробити такі висновки.

1. Найбільшого ущільнення (на 20 %) можна досягти при поданні у графічнокодованому вигляді десяткових даних ($P_A = 10$, коефіцієнт ущільнення дорівнює 1,2).

2. При створенні графічнокодових позначок невеликої ємності (до 256 ГК-знаків; $s = 8$) для подання алфавітно-цифрових даних доцільно використовувати алфавіт потужністю $P_A = 72$; при цьому досягається ущільнення 12,5 %.

3. При побудові ГК-позначок ємністю близько 512 ($s = 9$) ГК-знаків її слід заповнювати як два ГК-слова з параметрами $P_\Omega = 256$, $P_A = 72$, оскільки заповнення її як одного слова з параметрами $P_\Omega = 512$, $P_A = 86$ дає низький показник ущільнення (лише 8,9 %).

4. Високі показники ущільнення досягаються при побудові ГК-позначок середньої ємності (близько 1024 ГК-знаків; $s = 10$), якщо використовувати алфавіт потужністю $P_A = 140$ (20 % – при поданні десяткових даних, 12 % – при поданні алфавітно-цифрових даних).

5. При створенні ГК-позначок ємністю близько 2048 ГК-знаків ($s = 11$) прийнятні показники ущільнення досягаються при поданні алфавітно-цифрових даних, якщо використовувати потужність алфавіту $P_A = 69$. Але при цьому дещо незручно працювати з десятковими послідовностями внаслідок великої довжини цифрових підпослідовностей при перетворенні "23" → "7".

6. Створення ГК-позначок великої ємності (близько 4096 ГК-знаків; $s = 12$) потребує використання алфавіту потужністю $P_A = 146$, що є зручним, оскільки ця потужність є вищою за потужність стандартного ASCII (128 символів), але при цьому ущільнення буде дещо меншим – 11 %.

Таким чином, з точки зору забезпечення максимуму ущільнення алфавітно-цифрових даних і зручності роботи з послідовностями символів перевагу слід віддати створенню ГК-позначок з такими параметрами: ємність до 1024 ГК-знаків; при $P_A = 140$ – коефіцієнт ущільнення 1,12; при $P_A = 10$ – коефіцієнт ущільнення 1,2.

Режими кодування даних

Кількість режимів кодування залежить від кількості локальних екстремумів коефіцієнта ущільнення при фіксованому s.

Оскільки при $s = 10$ локальні екстремуми коефіцієнта ущільнення досягаються при по-

тужностях алфавіту 10, 140 (див. рис. 3), то доцільно використовувати три режими для подання алфавітно-цифрових повідомлень у ГК-вигляді (рис. 4):

1) режим ASCII (“А”) – для підпослідовностей, утворених із символів розширеного ASCII;

2) текстовий режим (“Т”) – для підпослідовностей символів з алфавіту T_{140} потужністю $P_T = 140$;

3) цифровий режим (“Ц”) – для підпослідовностей, що складаються з десяткових цифр, $P_Ц = 10$.

Для переходу з одного режиму в інший використовують спеціальні кодослова (рис. 4). У нашому випадку необхідно використовувати три перемикачі, наприклад:

- $J_A = 1022$ – символ-перемикач (кодовектор) для переходу в режим ASCII з поточного режиму;

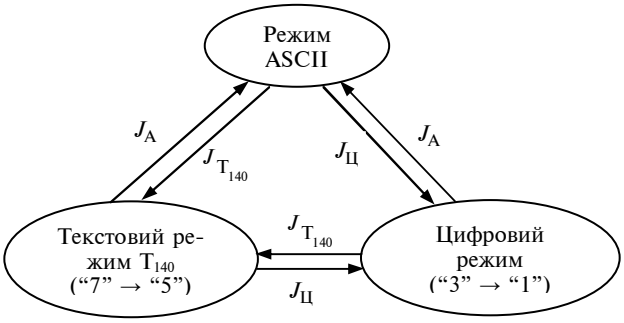


Рис. 4. Режими кодування алфавітно-цифрових послідовностей при $s = 10$ ($P_\Omega = 1024$, $P_{\Omega\ інф} = 1013$)

- $J_{T_{140}} = 1021$ – символ-перемикач для переходу в текстовий режим з поточного режиму;
- $J_Ц = 1020$ – символ-перемикач для переходу в цифровий режим з поточного режиму.

Алфавіт потужністю $P_{A(T)} = 140$ текстового режиму призначений для кодування кирилиці,

Таблиця 2. Структура алфавіту текстового режиму

Значення символа в текстовому режимі	Базовий набір		Додатковий набір 1		Додатковий набір 2		Додатковий набір 3	
	Латино-кириличний набір		Набір цифрових пар		Набір символів розширеного ASCII 0–127		Набір символів розширеного ASCII 128–255	
	Символ	Значення символа в ASCII	Символ	Значення символа в ASCII	Символ	Значення символа в ASCII	Символ	Значення символа в ASCII
0	A	128	00		NUL	0	А	128
1	Б	129	01		SOH	1	Б	129
2	B	130	02		STX	2	В	130
...	...	...	...	...	...	...	...	...
99	d	100	99		c	99	у	227
100	e	101	0		d	100	ф	228
101	f	102	1		e	101	х	229
...	...	...	...	...	...	...	...	...
109	n	110	9		m	109	э	237
...	...	...	...	...	...	...	...	...
127	5	53	R	82	DEL	127	FF	255
128	6	54	S	83	А	128	:	58
129	7	55	T	84	Б	129	"	34
130	8	56	U	85	В	130	(	40
131	9	57	V	86	Г	131	)	41
132	space	32	W	87	Д	132	space	32
133	,	44	X	88	Е	133	,	44
134	–	45	Y	89	Ж	134	–	45
135	.	46	Z	90	З	135	.	46
136	;	59	;	59	И	136	;	59
137	s_1		s_0		s_0		s_0	
138	s_2		s_2		s_1		s_1	
139	s_3		s_3		s_3		s_2	

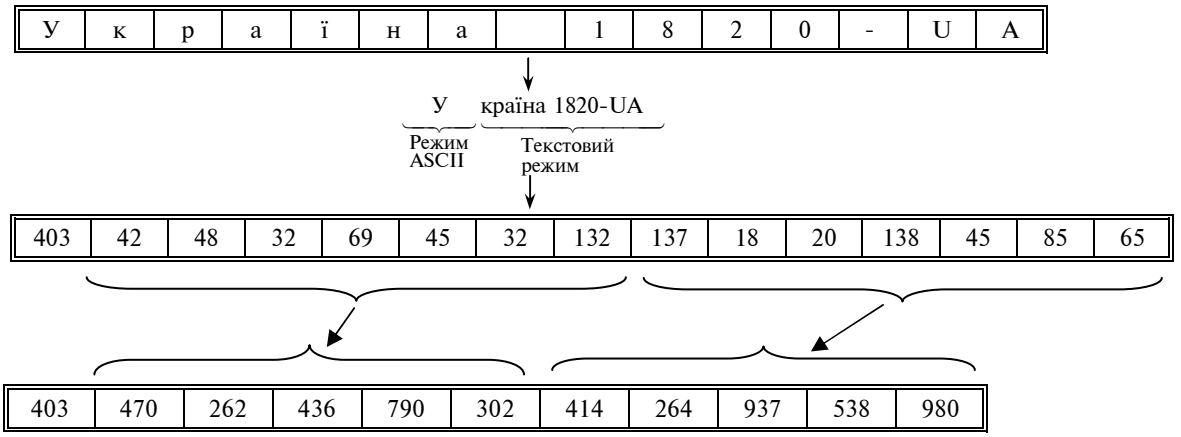


Рис. 5. Приклад кодування алфавітно-цифрової послідовності в текстовому режимі

латиниці, цифр і деяких службових символів (табл. 2). Алфавіт текстового режиму містить один базовий набір та три додаткових набори.

Базовий набір (140 символів) складається з великих і малих літер кирилиці (0–69), великих і малих літер латиниці (70–121), цифр (122–131), символів-зсувачів до додаткових наборів 1, 2, 3 (s_1, s_2, s_3).

Додатковий набір 1 (140 символів) складається з цифрових пар (00–99), десяткових цифр (100–109), великих літер латиниці (110–135), символів-зсувачів до базового набору (s_0), до додаткових наборів 2, 3 (s_2, s_3).

Додатковий набір 2 (140 символів) складається з 128 символів розширеного ASCII з кодами 0–127, деяких символів кирилиці (128–135), символів-зсувачів до базового набору (s_0), до додаткових наборів 1, 3 (s_1, s_3).

Додатковий набір 3 (140 символів) складається з 128 символів розширеного ASCII з кодами 128–255, деяких спеціальних символів (128–135), символів-зсувачів до базового набору (s_0), до додаткових наборів 1, 2 (s_1, s_2).

Нехай задано вхідну алфавітно-цифрову послідовність завдовжки 15 символів "Україна 1820-UA", якій при прямому поданні відповідають 120 біт.

Закодуємо цю послідовність (рис. 5).

Перший символ послідовності кодуємо в режимі ASCII. Символу "У" в ASCII відповідає код 147 (10010011). Двійковий код першого символа доповнимо зліва комбінацією 01, що свідчить про подальший перехід у текстовий режим, тому значення стартового кодовектора дорівнює 403 (0110010011).

Решту алфавітно-цифрових символів закодуємо з використанням алфавіту текстового режиму. Розіб'ємо їх на підпослідовності:

де s_1 – символ-зсувач з базового в додатковий набір 1 (його значення – 137 (див. табл. 2)), s_2 – символ-зсувач з додаткового набору 1 (набір цифрових пар) у додатковий набір 2 (його значення – 138).

Кожному символу поставимо у відповідність його значення з алфавіту текстового режиму.

Отриману послідовність числових значень символів

$$\underbrace{42\ 48\ 32\ 69\ 45\ 32\ 132}_{\text{"7"} \to \text{"5"}}\ \underbrace{137\ 18\ 20\ 138\ 45\ 85\ 65}_{\text{"7"} \to \text{"5"}}$$

розбиваємо на групи по 7 значень і для кожної з таких груп застосовуємо перетворення "7" → → "5" (семи значенням з системи числення з основою 140 (потужність алфавіту текстового режиму) ставимо у відповідність 5 значень з системи числення з основою $P_{\Omega\ \text{інф}} = 1013$). Перетворення "7" → "5" реалізуємо як перетворення числа з системи числення з основою 140 на число з системи числення з основою 1013.

Розглянемо перетворення для першої групи з семи значень. Будуємо поліном [2, 3]:

$$S(x) = a_6 x^6 + a_5 x^5 + a_4 x^4 + a_3 x^3 + a_2 x^2 + a_1 x + a_0,$$

де 140 – основа вхідної системи числення, коефіцієнти $\{a_6, a_5, a_4, a_3, a_2, a_1, a_0\}$ – група з семи символів {42, 48, 32, 69, 45, 132, 137}, що підлягає перетворенню.

Обчислюємо значення полінома:

$$S = 42 \cdot 140^6 + 48 \cdot 140^5 + 32 \cdot 140^4 + 69 \cdot 140^3 +$$

$$+ 45 \cdot 140^2 + 32 \cdot 140 + 132 = 318834550542612.$$

Результатом перетворення буде група з 5 символів у системі числення з основою 1013, тобто $\{r_4, r_3, r_2, r_1, r_0\}$.

Знаходимо кодослово r_4 як остачу від ділення значення полінома S на основу системи числення 1013 та проміжне значення S_4 як цілу частину від ділення S на 1013:

$r_4 = S \mod 1013 = 470$ (старше кодослово r_4),

$S_4 = S \operatorname{div} 1013 = 314742892934$ (проміжне значення),

де $\alpha \bmod \beta$ – остача від ділення α на β, $\alpha \operatorname{div} \beta$ – ціла частина від ділення α на β.

Виконуючи аналогічні операції над черговим проміжним значенням S_4 знаходимо решту значень – r_3, r_2, r_1, r_0:

$r_3 = S_4 \mod 1013 = 262;$

$S_3 = S_4 \operatorname{div} 1013 = 310703744;$

$r_2 = S_3 \mod 1013 = 436;$

$S_2 = S_3 \operatorname{div} 1013 = 306716;$

$r_1 = S_2 \mod 1013 = 790;$

$r_0 = S_2 \operatorname{div} 1013 = 302.$

Тобто

42 48 32 69 45 32 132 → 470 262 436 790 302.

Для другої групи з 7 символів аналогічно отримаємо (див. рис. 5):

137 18 20 138 45 85 65 → 414 264 937 538 980.

Остаточно вхідній послідовності “Україна 1820-UA” відповідають 11 кодослів:

403 470 262 436 790 302 414 264 937 538 980,

загальна довжина яких становить 110 біт (рис. 5).

Аналіз потоку алфавітно-цифрових даних

З метою підготовки даних до подання у графічнокодованому вигляді необхідно розробити аналізатор потоку вхідних алфавітно-цифрових символів, що є символами ASCII (комп’ютерного алфавіту). Робота аналізатора має забезпечувати взаємозв’язок режимів кодування даних відповідно до рис. 4.

Аналізатор потоку алфавітно-цифрових даних здійснює перевірку типу символів вхідної послідовності та її розбиття, якщо це необхідно, на підпослідовності символів з розставлянням відповідних символів-перемикачів до того чи іншого режиму кодування з метою досягнення якнайвищого ступеня ущільнення даних.

Кожну з визначених підпослідовностей кодують за допомогою одного з трьох режимів кодування даних: режиму ASCII, текстового режиму, цифрового режиму. Перший символ будь-якої вхідної послідовності завжди кодується в режимі ASCII. Вибір схеми кодування даних для кожної підпослідовності відбувається таким чином, щоб символи даної підпослідовності були закодовані якнайменшою кількістю кодослів.

Перетворення вхідної послідовності слід здійснювати у два проходи.

1. Проаналізувати вхідну послідовність посимвольно зліва направо і розбити її на підпослідовності суміжних символів, до яких входять лише символи алфавіту з одного режиму. Перед черговою підпослідовністю вставити відповідний символ-перемикач режиму.

2. Кожну отриману підпослідовність символів обробити за правилами відповідного режиму та перетворити на послідовність кодовекторів.

Структурний метод підвищення інформаційної щільності графічних кодів

Розглянуту послідовність перетворень і дій можна узагальнити як структурний метод підвищення інформаційної щільності графічних кодів. Нижче наведено узагальнене поетапне описання цього методу.

1. Якщо необхідно створювати ГК-позначки ємністю V ГК-знаків, то символіка Ω графічного коду повинна мати потужність $P_\Omega = 2^s$, де $s \geq]\log_2 V[$.

2. Вибираємо необхідну кількість службових кодовекторів $P_{\Omega_{\text{сл}}}$. Тоді кількість інформаційних кодовекторів становить $P_{\Omega_{\text{інф}}} = P_\Omega - P_{\Omega_{\text{сл}}}$.

3. Шукаємо локальні екстремуми коефіцієнта ущільнення $U_s^{(P_{\Omega_{\text{інф}}})}$. Для цього розв’язує-

мо систему (5) при відомому $P_{\Omega_{\text{інф}}}$. Розв'язками (5) є множина значень потужності P_{A} алфавітів, при яких $U_s^{(P_{\Omega_{\text{інф}}})} > 1$, та тип перетворення "$n$" → "$m$", при якому для кожного P_{A} досягається локальний екстремум $U_s^{(P_{\Omega_{\text{інф}}})}$.

4. Знаходимо кількість режимів кодування алфавітно-цифрових послідовностей: $R = L + 1$, де L – кількість вибраних для практичного застосування локальних екстремумів коефіцієнта ущільнення $U_s^{(P_{\Omega_{\text{інф}}})}$ (обов'язковим додатковим режимом кодування є режим ASCII). Визначаємо правила переходу між режимами.

5. Для кожного режиму кодування формуємо алфавіти, поділені на набори символів, та визначаємо правила переходу між наборами та між режимами.

6. Формуємо правила розбиття вхідної алфавітно-цифрової послідовності на підпослідовності суміжних символів, до яких входять лише символи алфавіту з одного режиму.

7. Кожну отриману підпослідовність символів обробляємо за правилами відповідного режиму та перетворюємо на послідовність кодовекторів.

Висновки

Ущільнення алфавітно-цифрової послідовності є обов'язковим етапом при її поданні у вигляді графічного коду. Можливість ущільнення ґрунтується на перетворенні даних з одного алфавіту в інший – з алфавіту символів в алфавіт ГК-знаків графічного коду. Ущільнення даних досягається використанням кількох алфавітів символів – алфавіту десяткових цифр, текстових символів, розширеного ASCII. Це означає, що вхідну алфавітно-цифрову послідовність варто поділяти на відповідні підпослідовності символів – цифрові підпослідовності, підпослідовності текстових символів, підпослідовності символів розширеного ASCII, і в межах кожної підпослідовності виконувати відповідне переведення символів – з алфавіту поточної підпослідовності в алфавіт ГК-знаків.

Коефіцієнт ущільнення даних, якщо його розглядати як функцію потужності використовуваного алфавіту символів, може мати кілька локальних екстремумів. Саме в точках екстремумів і досягається максимум ущільнення.

При бітовому поданні даних у ГК-позначці найкращі показники ущільнення (на 20 %) досягаються у випадку подання в графічнокодованому вигляді цифрових даних, для текстових послідовностей ущільнення можливе на 12 %.

Запропонований метод ущільнення даних можна використовувати для довільного алфавіту символьних послідовностей.

Подальші дослідження слід зосередити на з'ясуванні особливостей ущільнення даних у випадку побудови багатоколірних графічних кодів.

1. *Арманд В.А., Железнов В.В.* Штриховые коды в системах обработки информации. – М.: Радио и связь, 1989. – 92 с.
2. *Питерсон У., Уэлдон Э.* Коды, исправляющие ошибки. – М.: Мир, 1976. – 594 с.
3. *Цымбал В.П.* Теория информации и кодирование. – К.: Вища шк., 1982. – 304 с.
4. *Elfner R.W.* Bar Code Printing on Shipping Containers. – Helmers Publishing, 1994. – 248 p.
5. *Palmer R.C.* The Bar Code Book: Reading, Printing & Specification of Bar Code Symbols. – Helmers Publishing, 1990. – 320 p.
6. *Williams T.* Data Matrix Is Lazerlight Systems. – Computype Intern. Inc., 1990. – 22 p.

Рекомендована Радою
факультету прикладної математики
НТУУ "КПІ"

Надійшла до редакції
14 лютого 2011 року