

(нехай це буде p), що $p < 2(n - k)$. Причому, або $p > n$, і доведення закінчене, або $p \leq n$. Припустимо, що $p \leq n$. Тоді або p не ділить n , або p ділить n . Розглянемо перший випадок. Тоді $n - p = a$, $2n - p = q + 2k$. Віднявши і додавши ці дві рівності, знайдемо, що $n = q - a + 2k$, а також $3n - 2p = q + a + 2k$. Звідси випливає, що виконується рівність $-a^2 = 3n^2 - 2(p + 4k)n + 4k(p + k) - q^2$ при кожному натуральному n . Ліва частина останньої рівності – повний квадрат для кожного n . Тоді дискримінант правої її частини дорівнює нулеві. Це означає, що має місце наступна рівність: $(p - 2k)^2 + 3q^2 = 0$, що неможливо, оскільки $p \neq 0$ і $q \neq 0$. Отже, якщо $p \leq n$, то p ділить n . Нехай $n = mp$. Тоді виконується рівність $(2m - 1)p - 2k = q$, звідки в силу того, що $q > 2$ – просте, випливає що $(p, k) = 1$. Нехай $p \leq n/2$. Тоді $n - 2p = a$, $2n - p = q + 2k$. З цих рівностей знайдемо, що $n + p = q + 2k - a$, а також, що $3n - 3p = q + 2k + a$. Звідси випливає, що при кожному натуральному n виконується: $-a^2 = 3n^2 - [3p^2 + (q + 2k)^2]$. Ліва частина останньої рівності – повний квадрат для кожного n . Тоді дискримінант правої її частини дорівнює нулеві. Це означає, що має місце наступна рівність: $(q + 2k)^2 + 3p^2 = 0$, що неможливо, оскільки $p \neq 0$ і $q > 0$. Отже, $p \geq n/2$, тобто $n \leq 2p$. Розділивши останню нерівність на p , отримаємо, що $m \leq 2$. При $m = 1$ $p - q = 2k$, $n = p$ і теорема 12 доведена. А оскільки $4p - 2k < 4p - 2n/2 = 2p$, тобто $p < k < n/2$, що суперечить припущенню, то випадок $m = 2n = 2p$ неможливий. Отримані суперечності доводять, що нерівність $p \leq n$ неможлива. Теорема 12 доведена.

1. Sierpin'ski W. Elementary theory of numbers. Warsawa, 1964, Pan'stwowe wydawnictwo naukowe, С.335.
2. Прахар К. Распределение простых чисел. М., 1967

УДК 681. 513

РЕАЛІЗАЦІЯ СТРУКТУРНОГО АЛГОРИТМУ РОЗПІЗНАВАННЯ ОБРАЗІВ НА ОДНОРІДНИХ ОБЧИСЛЮВАЛЬНИХ СИСТЕМАХ

© А. Худий

Державний науково-дослідний інститут інформаційної інфраструктури

Запропоновано реалізацію структурного алгоритму розпізнавання образів на ООС (однорідних обчислювальних системах) на прикладі зображення у вигляді матриці розмірністю 256×256.

A reduced realization of structure algorithm of recognition of images on the homogenous calculus systems.

Незалежно від підходу або методу, який застосовується для розпізнавання образів, задача розпізнавання буде розв'язана повністю лише тоді, коли алгоритм розпізнавання інваріантний відносно афінних перетворень образів у полі аналізу. Чутливість алгоритму до паралельних переносів, поворотів і масштабних змін досліджуваних образів призводить до ускладнення процесу розпізнавання і до помилок при прийнятті рішення.

Відомо [1], що в загальному випадку усунення залежності алгоритму розпізнавання від афінних перетворень досягається двома шляхами: механічним або електричним переміщенням зображень відносно еталона або навпаки; або аналітичним визначенням ознак образів, інваріантних лінійному переміщенню, гомотетії та повороту. Першому методу притаманні певні недоліки, серед яких виділимо такі: низька достовірність і обмеженість класів образів, які розпізнаються; суттєве збільшення часу на прийняття кінцевого рішення; ускладнення апаратних затрат. Все це загалом не дозволяє розв'язувати задачі розпізнавання образів у реальному часі.

Ознаки, які є інваріантними до афінних перетворень, є структурними характеристиками: $T_s(x)$ і $L_s(x)$ [2]. В основу структурного алгоритму розпізнавання образів (САР) покладена властивість неперекриття структурних траєкторій і областей класів образів, які розпізнаються [3].

Нехай маємо деяке зображення (матрицю) розмірністю 256×256 . Відомо, що

$$a_{ij} = \begin{cases} 1, & (i = \overline{1, 256}, j = \overline{1, 256}) \\ 0, & \end{cases}$$

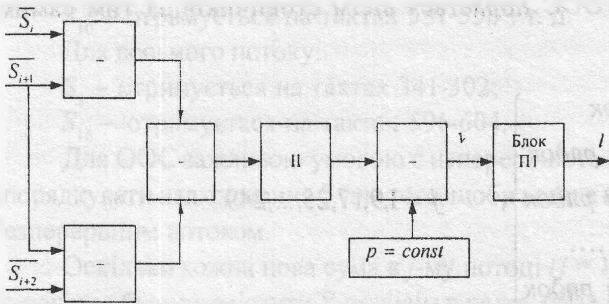


Рис. 1. Послідовність виконання дій на другому етапі реалізації

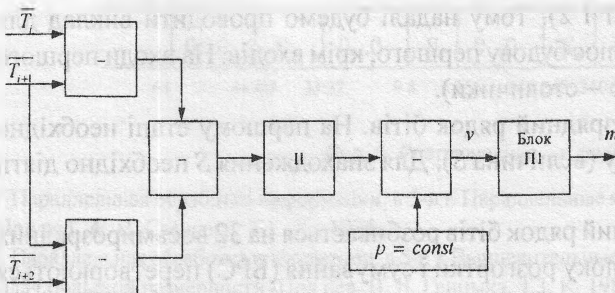


Рис. 2. Послідовність виконання дій на четвертому етапі реалізації

Необхідно на ООС реалізувати таку послідовність дій:

1. Знайти суму кожного рядка

$$S_i = \sum_{j=1}^{256} a_{ij}, \quad i = \overline{1, 256}.$$

2. Після знаходження суми рядків необхідно виконати такі дії (див. рис. 1).

Блок ПІ обчислює кількість виконання умови $v - p > 0$, де p – деяка константа.

Нехай l – результат роботи блока ПІ.

3. Знайти суму кожного стовпчика

$$T_j = \sum_{i=1}^{256} a_{ij}, \quad j = \overline{1, 256}.$$

4. Після знаходження суми всіх стовпчиків необхідно виконати такі дії (див. рис. 2).

Блок ПІ обчислює кількість виконання умови $v - p > 0$, де p – деяка константа.

Нехай m – результат роботи блока ПІ.

5. Наостанньому етапі необхідно виконати такі дії (див. рис. 3).

Для зручності кожен рядок (стовпчик) зображення можна інтерпретувати як 256-ти розрядний рядок бітів. Припустимо, що хоча б один біт цього рядка дорівнює нулю.

Як відомо, кожен блок ООС (96×46) має вісім входів (вісім виходів). Для максимального завантаження будемо вважати, що на блок одночасно подається вісім рядків, тобто:

$$\left. \begin{array}{l} \text{вхід } 1 - i - \text{й рядок} \\ \text{вхід } 2 - (i + 1) - \text{й рядок} \\ \text{вхід } 3 - (i + 2) - \text{й рядок} \\ \dots \\ \text{вхід } 8 - (i + 7) - \text{й рядок} \end{array} \right\}, \quad i = 1, 9, 17, 25, \dots, 249.$$

Одночасно на другий блок ООС подається вісім стовпчиків за тим самим принципом:

$$\left. \begin{array}{l} \text{вхід } 1 - j - \text{й стовпчик} \\ \text{вхід } 2 - (j + 1) - \text{й стовпчик} \\ \text{вхід } 3 - (j + 2) - \text{й стовпчик} \\ \dots \\ \text{вхід } 8 - (j + 7) - \text{й стовпчик} \end{array} \right\}, \quad j = 1, 9, 17, 25, \dots, 249.$$

Перший і другий кроки алгоритму будуть виконуватися в першому блоку ООС, а третій і четвертий – в другому блоку ООС. По суті, в обох блоках необхідно виконати абсолютно тотожні дії (див. рис. 1 і 2). Тому надалі будемо проводити виклад для першого блока (другий просто дублює будову першого, крім входів. На входи першого подаються рядки, на входи другого – стовпчики).

Кожен рядок – це 256-ти розрядний рядок бітів. На першому етапі необхідно знайти суму одиниць в цьому рядку (величина S). Для знаходження S необхідно діяти у два етапи.

Перший етап. 256-ти розрядний рядок бітів розбивається на 32 восьмирозрядних підрядки бітів, які за допомогою блоку розгортки і сумування (БРС) перетворюються у числа R_i , $i = 1, 32$, де R_i – кількість одиниць в i -му восьмирозрядному підрядку бітів (див. рис. 4).

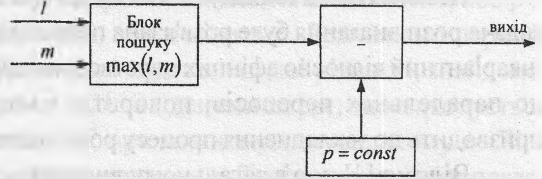


Рис. 3. Послідовність виконання дій на останньому етапі реалізації

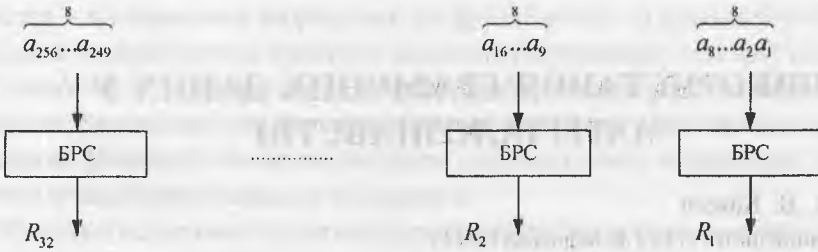


Рис. 4. Схема розкладу 256-ти розрядного рядка бітів на 32 восьмирозрядних підрядки бітів

Зауважимо, що 256-ти розрядний рядок бітів подається на один блок розгортки і сумування, тобто сам блок ділить цей рядок на підрядки по вісім бітів. Очевидно, що в блоку ООС необхідно мати вісім блоків БРС (для кожного рядка).

Другий етап. Безпосередньо знаходимо величину

$$S = \sum_{i=1}^{32} R_i$$

Нехай S_i – сума одиниць i -го рядка ($i = 1, 256$). При отриманні S_i утворюються розриви в потоках, тобто видача S_i отримується на тактах 309-316, S_9 – на тактах 565-572 і т. д.

Аналогічно для другого потоку:

S_2 – отримується на тактах 295-302;

S_{10} – отримується на тактах 551-558 і т. д.

Для восьмого потоку:

S_8 – отримується на тактах 341-302;

S_{16} – отримується на тактах 596-604.

Для ООС важливою умовою є неперервність потоків. У зв'язку з цим необхідно впорядкувати надходження S_j для того, щоби вони в подальшому надходили на обробку безперервним потоком.

Оскільки кожна нова сума в j -му потоці ($j = 1, 8$) отримується через 256 тактів, то логічно буде розмістити S_j по вісім в рядку бітів розмірністю 256 (рис. 5).

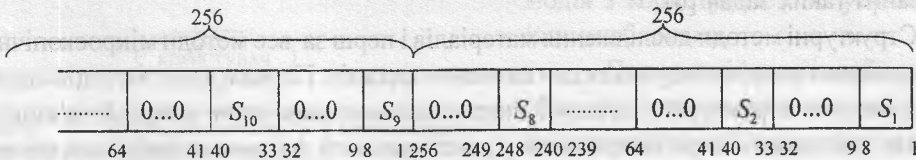


Рис. 5. Розташування в рядку бітів

1. Параллельная обработка информации: в 5-и т. Параллельные методы и средства распознавания образов /Под ред. А. Н. Свенсона. Т.1 . К., 1985.
2. Параллельная обработка информации: в 5-и т. Вычислительные системы, структуры и среды для решения задач большой размерности /Под ред. В. В. Грицыка. Т.3. К, 1986.
3. Худий А.М. Структурний алгоритм розпізнавання образів, Вісник ДУ "Львівська політехніка", 1999р., № 364, с.333-340.