

Модель відображається мовою XML відповідно до підходів, використаних у [7, 8] та є готовою для опрацювання інтерпретатором. Перед використанням розроблену модель тестують на відповідність початковим вимогам та варіантам використання.

### Висновки

Запропонований у роботі підхід до побудови виконувальних моделей сервісів інформаційних систем дасть змогу створювати системи, які порівняно з існуючими швидше адаптуються до змін у навколишньому середовищі за зменшення видатків на проектування та супровід.

1. Balmelli L, Brown D, Cantor M, Mott M. *Model-driven systems development [Text]* / Balmelli L, Brown D, Cantor M, Mott M // *IBM Systems Journal* .- 2006.-, vol 45, # 3. 2. Буров Є.В. Система формальних специфікацій для проектування розподілених інформаційних систем [Текст]/Буров Є.В.// Вісник держуніверситету 'Львівська політехніка' 'Інформаційні системи та мережі'.- 2000.- № 406. 3. MDA Distilled. Principles of Model Driven Architecture, Stephen Mellor, Kendall Scott, Axel Uhl, Dirk Weise.[Text].- Addison-Wesley Professional.- 2004. 5. Stephen J. Mellor, Marc J. Balcer. *Executable UML: A foundation for model-driven architecture [Text]*./Mellor S, Balcer M.- Addison Wesley,2002.- 416p.-ISBN 0-201-74804-5 6. OMG. *Semantic profile for UML[Electronic resource]*. - Mode of access: URL: <http://www.omg.org/cgi-bin/doc?formal/03-03-01>. 7. Avesha Malik. *Design XML schemas using UML. [Electronic resource]*.- Mode of access: URL: <http://www.ibm.com/developerworks/library/x-umlchem/>.-Title from the screen. 8. Benoit Marshal. *Working XML: UML,XMI and code generation. [Electronic resource]*. – Mode of access: URL: <http://www.ibm.com/developerworks/xml/library/x-wxxm23/>.- Title from the screen.

УДК 622.02.658.284

Б. Демида

Національний університет "Львівська політехніка",  
кафедра автоматизованих систем управління

## МЕТОД РОЗПІЗНАВАННЯ ШТРИХ-КОВОЇ ІНФОРМАЦІЇ (Ч. II)

© Демида Б., 2009

**Розглянуто метод автоматичного розпізнавання штрих-кової інформації зі звітних листків ідентифікації виробничих операцій виконаної роботи для системи автоматизації обліку на швейному виробництві.**

**This paper considers the method of automatic recognition of bar code information from reports of identification of production operations of the performed work for the account automation system in sewing industry.**

### 1. Загальний опис алгоритму

У першій частині статті [1] описано алгоритм пошуку та локалізації штрих-кодових об'єктів із застосуванням таких кроків:

- сегментація графічного образу листка формату А4 з наклеєними мітками на окремі графічні елементи з перевіркою наявності на цих елементах бар-кодів за допомогою функції фільтрації випадкових величин.

У цій статті опишемо:

1. Алгоритм визначення граничних областей, графічного центру та кута повороту елемента бар-коду. Для реалізації цього алгоритму застосуємо:

- періодичне n-разове сканування по вертикалі локалізованої області з певним кроком із застосуванням методу пошуку;

- визначення графічного центру в кожному проході сканування та побудова з цих графічних центрів лінії нахилу елемента з бар-кодом;
  - визначення кута нахилу елемента відносно лінії сканування графічного образу листа;
2. Контрольний приклад роботи алгоритму пошуку та локалізації штрих-кодів
  3. Алгоритм виділення структурних елементів штрих-кодів. Для реалізації цього алгоритму застосуємо:
    - метод виділення вектора яскравостей уздовж прямої (враховуємо кут нахилу елемента), що перетинає всі штрихи коду з переведенням нечітких значень ширини елементів символіки (штрихів і пропусків) в числа, кратні модулю (елементу мінімальної ширини) штрихового коду;
  4. Алгоритм декодування штрих-кодів:
    - табличне порівняння отриманого значення із стандартом Бар-коду типу Code 128;
    - процедура визначення наявності у коді помилки;
    - декодування коду з розворотом на 180 град.;
    - прийняття рішення про розпізнавання Бар-коду.

## 2. Алгоритм визначення орієнтації штрих-кодів

Робота алгоритму полягає в знаходженні рівняння прямої, що перетинає всі елементи символіки штрих-коду. У цьому алгоритмі було вирішено будувати пряму, опираючись на скелет фігури, створеної елементами штрих-коду. Операція знаходження скелета фігури називається скелетизацією. Суть операції скелетизації полягає в керованій послідовній рівномірній ерозії об'єкта [2]. Ерозія продовжується доти, поки не залишаться лінії завширшки в одну точку. Ці лінії через алгоритм і ерозію є рівновіддаленими від двох ближніх до них сторін об'єкта, і їх ширина становить одну точку (рис. 1).



Рис. 1. Скелетизація фігур, створених елементами штрих-коду

З огляду на те, що скелет складається з серединних ліній, пряма, що проходить через вузли скелета, перетне всі елементи штрихового коду. Отримане рівняння прямої усередині області необхідно перетворити до рівняння в початковому зображенні:

$$\begin{aligned} x = k_x \cdot t + x_0 &\Rightarrow x = k_x \cdot t + (x_0 + x_B) \cdot SCALE \\ y = k_y \cdot t + y_0 &\Rightarrow y = k_y \cdot t + (y_0 + y_B) \cdot SCALE \end{aligned} \quad (1)$$

де  $k_x$ ,  $k_y$ ,  $x_0$  і  $y_0$  – коефіцієнти параметричного рівняння всередині прямокутної області;  $x_B$ ,  $y_B$  – координати початку області в зображенні з низьким розширенням;  $SCALE$  – лінійний коефіцієнт перетворення між системою координат початкового зображення і отриманих – меншого розширення.

Приклад виділення відрізка прямої, що перетинає всі елементи штрих-коду, показаний на рис. 2, а на рис. 3 зображені яскравості уздовж декількох відрізків паралельних виділеному. Описаний алгоритм спочатку задумувався для виявлення лінійних одновимірних штрих-кодів. Але він може бути з успіхом застосований для локалізації багаторядкових і модульованих по висоті символік, описаних в [3], оскільки в них знову ж таки існує просторова монотонність в одному напрямі. Незначна модифікація алгоритму, скерована саме на пошук найінтенсивніших, з погляду зміни контрасту, областей, дає змогу виявляти 2-мірні символіки, також описані в [3]. Очевидно, операція фільтрації з представленими в табл. 1 [1] фільтрами може бути основною при виявленні багатьох типів символік.



Рис. 2. Знайдений відрізок, що перетинає всі елементи символіки



Рис. 3. Значення яскравостей уздовж декількох суміжних відрізків, паралельних знайденому

### 3. Контрольний приклад роботи алгоритму пошуку та локалізації штрих кодів об'єктів

На вхід алгоритму подавалися штрих-коди з різними значеннями ширини найвужчого елемента (модуля), а також з різною амплітудою зміни яскравості уздовж всієї символіки. Один і той самий штрих-код подавався під різними кутами, що рівномірно змінюються (з кроком  $10^0$ ) в площині зображення. Бралось максимальне значення яскравості однієї і тієї самої точки штрих-коду з двох зображень, отриманих після операції поелементної різниці зображень. Були побудовані залежності яскравостей точок (розташованих в межах зображення штрих-коду) від кута повороту символіки. Результати роботи алгоритму показані на рис. 4:

- поріг бінаризації зображення після операції обчислення поелементної різниці, вибраний при розробленні алгоритму локалізації штрих-коду, дорівнює 0,02 і зображений на рисунку (пряма 4);
- штрих-код з шириною модуля 2,5 точки зображення і амплітудою зміни яскравості 0,5 від динамічного діапазону (крива 1);
- штрих-код з шириною модуля 5 точок зображення і амплітудою зміни яскравості 0,5 від динамічного діапазону (крива 2);
- штрих-код з шириною модуля 2,5 точки зображення і амплітудою зміни яскравості 0,25 від динамічного діапазону (крива 3);
- текст з товщиною ліній близько 2,5 точок зображення і амплітудою зміни яскравості 0,5 від динамічного діапазону (крива 5).

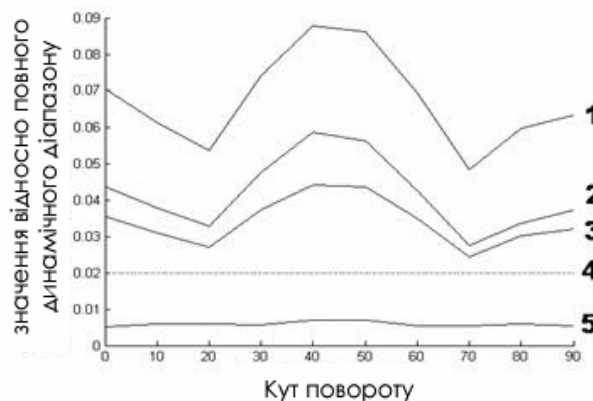


Рис. 4. Залежність яскравості точки об'єкта після операції поелементної різниці залежно від кута повороту в площині зображення

Всі криві, що належать до зображення штрих-коду, виявилися вищими за порогове значення, водночас, такий контрастний об'єкт, як текст виявився нижчим за поріг і випав з розгляду. Операції, що йдуть після операції поелементної різниці алгоритму локалізації штрих-коду, не потребують експериментальних досліджень, оскільки їхні дії цілком детерміновані. Скелет прямокутної фігури (символіки штрих-коду) має два вузли (у випадку, якщо він не пошкоджений і не загороджений іншими об'єктами), і пряма, що проходить через них, визначається однозначно.

#### 4. Алгоритм виділення структурних елементів штрих-кодових об'єктів

Відповідно до теореми Котельникова, задамося низьким мінімальним розширенням – 2 точки на модуль і, отже, дамо можливість більшій кількості точок знаходитися в діапазоні граничних яскравостей. Точка, що знаходиться в граничному діапазоні, може ототожнитися як з пропуском, так і зі штрихом, залежно від положення значення яскравості в граничній області і від вибору порогового значення (див. рис. 5). Це, своєю чергою, може спричинити неправильне визначення меж штрихів і пропусків і привести до помилки декодування штрих-коду. Отже, для поставленого завдання застосування одного порогового значення недостатньо. Інформація про точне положення межі модуля з точністю вищою, ніж роздільна здатність зображення, так само міститься в яскравості граничної точки. Тому необхідно використовувати методи, що дають змогу спиратися на просторове положення і значення яскравості граничної точки для уточнення межі між штрихом і пропуском.

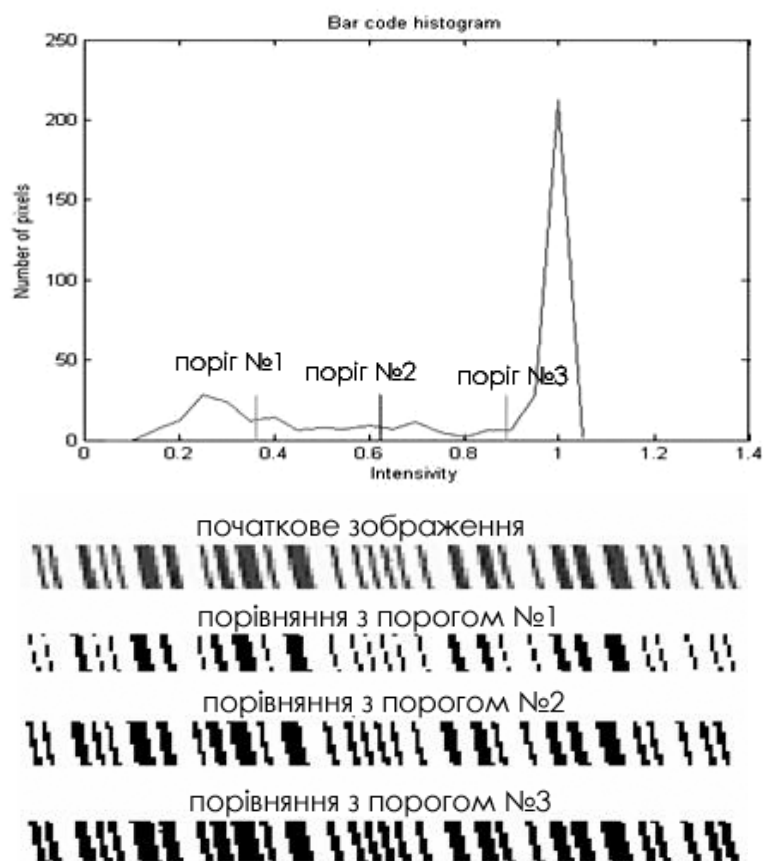


Рис. 5. Застосування порогової операції з трьома різними значеннями до початкового зображення штрих-коду з його низьким просторовим розширенням

На рис. 6 показана типова картина проєкції межі між штрихом і пропуском на світлочутливі елементи датчика зображення. Видно, що точки (1,3), (2,3), (2,2), (2,1) і (3,1) потрапили на межу розділу яскравості між штрихом і пропуском. Ці точки за період експозиції CCD накопичають деяке

середнє значення яскравості і потраплять в діапазон граничних яркостей гістограми розподілу значень яскравості.

Нехай в діапазоні граничних яркостей деяким чином вибрано три пороги яскравості, причому:  
 Порог №1 < Порог №2 < Порог №3

Після застосування 3-х операцій порівняння залежно від яскравості пікселя йому для кожного значення порогу буде привласнена або 1, або 0.

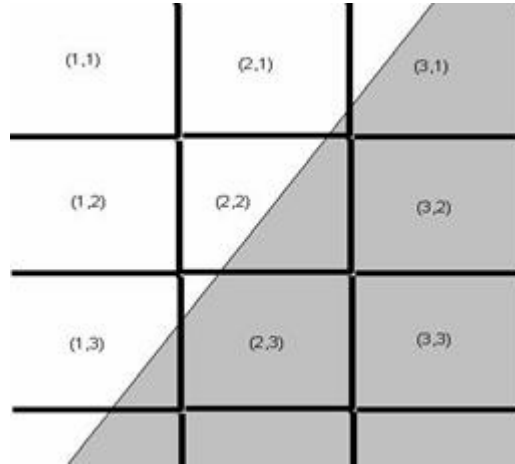


Рис. 6. Засвітка елементів світлочутливої матриці під час проходження по їх площі межі розділу освітленості штриха і пропуску

Для кожної точки початкового, сірого зображення (назвемо її материнською) у відповідність поставимо 3 точки отриманого зображення (назвемо їх нащадками). Нащадки містять набір яркостей (одиниць і нулів), отриманих після операцій порівняння з порогами материнської точки. При цьому однакові значення згруповані за розташуванням, наприклад, 001 або 011, тоді, як 010 або 101 – неможливі. Правила перетворення наочно зображені на рис. 7.

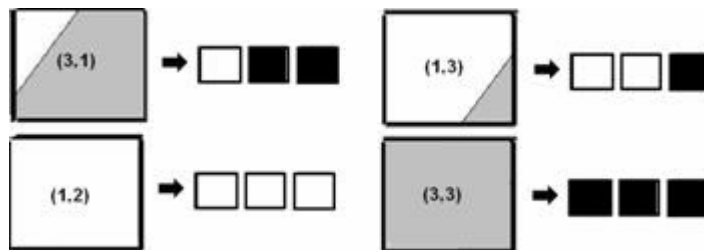


Рис. 7. Правила перетворення яркостей точок в бінарне зображення



Рис. 8. Приклад підвищення роздільної здатності на межі штрихів і пропусків

На рис. 8 показаний приклад роботи цього методу трьох порогів. Гістограма аналізованого зображення, а також результати його перетворення з трьома значеннями порогів, наведені на рис. 9.

Наступний етап розпізнавання штрих-коду полягає у збиранні статистики щодо ширині штрихів і пропусків. Операція полягає в побудові гістограми кількості штрихів і пропусків залежно від їх ширини (див. рис. 9). Для зменшення помилок, пов'язаних з дискретністю зображення, а також дефектами нанесення штрих-коду, використовується підсумовування (накопичення) значень ширини одних і тих самих елементів на різних рядках зображення. При правильному виборі порогів на такій гістограмі спостерігатиметься сукупність кривих, що нагадують криві нормального закону розподілу. Математичні сподівання цих центрів розміщуються один від одного на відстані, яка дорівнює ширині одного модуля.

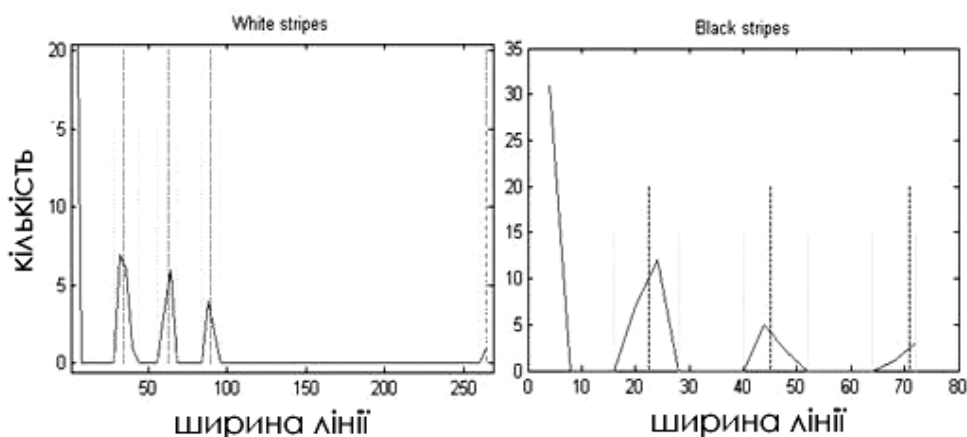


Рис. 9. Гістограма розподілу ширини пропусків (справа) і штрихів (зліва)

Отже, гістограми ширини штрихів і пропусків виявляються корисними для оцінювання правильності вибору порогів. Крім того, їх використання дає змогу визначити діапазони ширини штрихів і пропусків, яким необхідно приписати одну і ту саму кількість модулів ширини.

Отримані діапазони ширини можна перевести в одиниці вимірювання, кратні цілій кількості ширини модуля, що відповідає елементу штрих-коду мінімальної ширини. Вектор таких значень передається для використання в алгоритмі декодування штрих-коду.

## 5. Алгоритм декодування штрих-коду

Оскільки все, що потрібно зробити – це перетворити модульні (цілочислові) значення пропусків і штрихів на їхні символні еквіваленти відповідно до стандарту (до таблиці), то цей алгоритм є доволі простим. Єдине, що потрібно ще додатково реалізувати – це здійснити перевірку на правильність зчитування за виразом 6.

$$N_{start} + \sum_{i=1}^k (n_i \times N_i), \quad (6)$$

де  $N_{start}$  – числове значення знаку;

$k$  – кількість знаків у кодовому рядку;

$n_i$  – номер позиції знаку в кодовому рядку, рахуючи з першого знаку, що йде за знаком "Старт";

$N_i$  – числове значення знаку в  $i$ -й позиції.

Таблиця декодування, що описує стандарт Code 128, взята з [4].

На рис. 10 представлена блок-схема реалізованого алгоритму декодування символік Code 128.

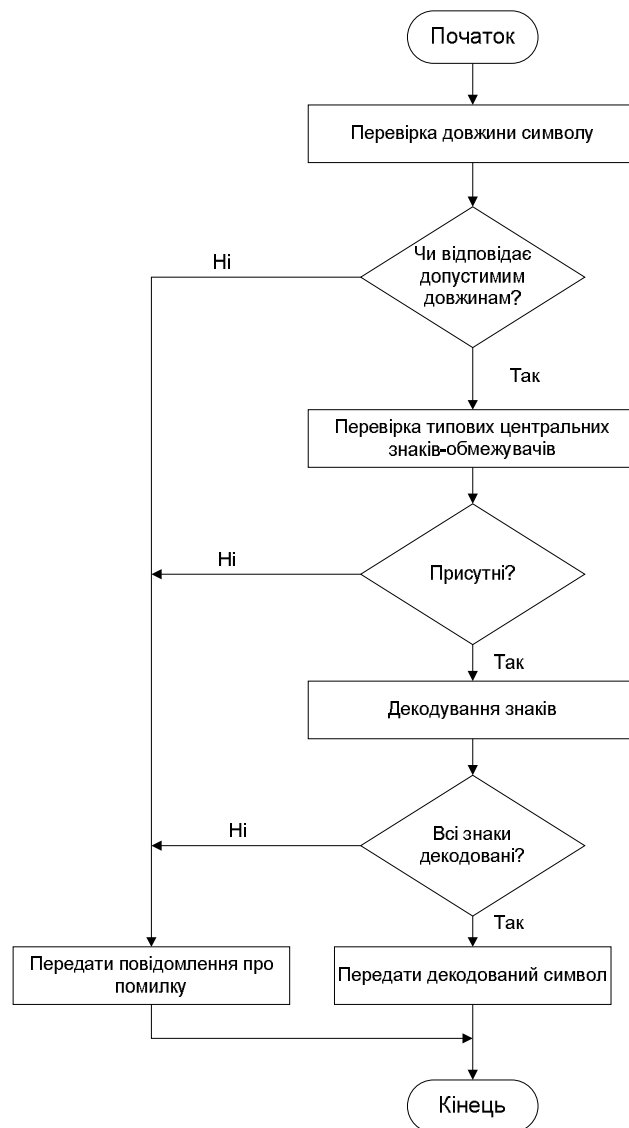


Рис. 10. Блок-схема алгоритму декодування символік Code 128

### Висновки

Розроблення алгоритму виділення структурних елементів штрих-коду передбачає невизначення розміру і положення вхідних даних. У алгоритмі також застосовуються ітераційні методи для визначення розташування штрихів і пропусків. Отже, об'єм використовуваної пам'яті і режим роботи алгоритму змінюються залежно від вхідних даних. Цей алгоритм повністю реалізується програмно.

Алгоритм декодування штрих-коду є алгоритмом верхнього рівня. Він отримує на вхід значно менший об'єм інформації, має велику безліч варіантів роботи залежно від типу штрих-коду.

Варто передбачити можливість нерозпізнаної штрихової символіки. В такому випадку оператор реальної системи повинен мати можливість ввести вручну нерозпізнану інформацію за допомогою ручного сканера і/або з клавіатури.

Алгоритм розпізнавання штрих кодової інформації (класи *MRecognize* та *MBarcode*) знайшов своє практичне застосування в системі автоматизації обліку на швейному виробництві [5].

1. Деміда Б. Метод розпізнавання штрих-кової інформації // Вісник Нац. ун-ту "Львівська полтехніка" "Комп'ютерні науки та інформаційні технології". – 2008. 2. Davies E.R. Machine Vision 3 Roger C. Palmer, The bar code book. Helmers Publishing, Inc. Formerly North American Technology, Inc., 1995. 4. <http://www.consumerinfo.org.ua>. Штрих-код. 5. Деміда Б., Горон М. Система автоматизації обліку на швейному виробництві // Вісник Нац. ун-ту "Львівська полтехніка" "Комп'ютерні науки та інформаційні технології". – 2008.– № 616.