

Б. Демида, Т. Кирик  
 Національний університет "Львівська політехніка",  
 кафедра автоматизованих систем управління

## МЕТОД РОЗПІЗНАВАННЯ ШТРИХ-КОДОВОЇ ІНФОРМАЦІЇ (Ч. I)

© Демида Б., Кирик Т., 2008

**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. Постановка проблеми та цілі

Для швидкого реагування на попит у швейній промисловості необхідно розробляти системи «малої» автоматизації обліку виконання замовлення. Приклад системи розглянуто у [1].

В основу автоматизації обліку такої системи покладено процедуру автоматичного сканування виконаної роботи зі звітних робочих листків операторів виробничих операцій і внесення цієї інформації в базу даних системи. Звітний робочий листок формується оператором методом перенесення клейких елементів (міток) з штрих-ковою інформацією операцій при виконанні тієї чи іншої роботи. Так виникло завдання автоматичного розпізнавання об'єктів зі штрих-ковою інформацією на растровому зображенні.

Штриховий код – це послідовність чорних та білих смужок, які представляють деяку інформацію у вигляді, зручному для зчитування технічними засобами. Штрихові коди використовуються в торгівлі, складському обліку, охоронних системах, на пошті, при обробці документів. В липні 2007 року штрих-коду "EAN" виповнилося 30 років. Ідея застосування лінійного штрих-коду була реалізована в 1948 році двома американськими студентами та ґрунтувалася на стандартному штрих-коді, який містив 13 штрихових цифр. Зараз застосування штрих-коду стало глобальним стандартом у всьому світі. Більш ніж п'ять мільярдів штрих-кодів сканується кожен день в усьому світі.

Штриховий Code128 є лінійним безперервним двонапрямленим контролепридатним кодом змінної довжини і дає змогу відобразити 128 знаків ASCII. Будова цього коду складається із зони стабілізації (1), що передує знаку "Старт", відповідного знаку "Старт" (2) підсистеми А, В або С, інформаційних знаків (3), зокрема і контрольного знаку (4), знаку "Стоп" (5) і зони стабілізації (6), що слідує за знаком "Стоп" (рис. 1) [2].



Рис. 1. Будова штрихової символіки стандарту Code128

Контрольне число – це залишок, отриманий від ділення на значення стартового символу підсистеми (за таблицею) значення виразу:

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

де  $N_{start}$  – числове значення знаку;  $k$  – кількість знаків в кодовій стрічці;  $n_i$  – номер позиції знаку у кодовій стрічці, рахуючи з першого знаку, що йде за знаком "Старт"; – числове значення знаку в  $i$ -й позиції.

У загальному випадку алгоритм розпізнавання штрих-кової інформації описаний, наприклад, у [3], і складається з трьох частини, як показано на рис. 2.

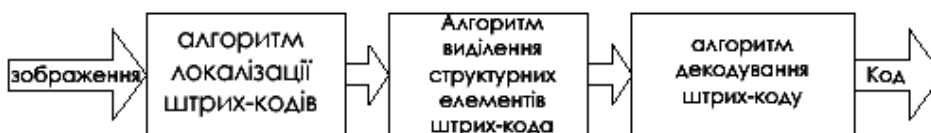


Рис. 2. Блок-схема алгоритму розпізнавання штрих-кодів

Алгоритм локалізації штрих-кодів містить частину, яка визначає ділянку зображення, що містить штрихову символіку з мінімальною висотою 5 мм. Усередині виділеної ділянки шукається пряма, що перетинає всі штрихи символіки і формує вектор яскравості уздовж цієї прямої. Отриманий вектор яскравості передається на алгоритм виділення структурних елементів, де здійснюється переклад нечітких значень ширини елементів символіки (штрихів і пропусків) в числа, кратні модулю (елементу мінімальної ширини) штрихового коду. На виході алгоритму виходить цілочисловий вектор значень ширини елементів. Алгоритм декодування розбирає структуру цілочисельного вектора і декодує закодовані символи відповідно до кодових таблиць.

Існує велика номенклатура різних за складністю та ціною пристроїв сканування штрих-кодів, побудованих на базі сигнального процесора [4].

Але, як вже було сказано вище, задача полягає в автоматичному розпізнаванні штрих-кодових об'єктів з растрових зображень звітів працівників, зчитаних сканером з автоподачею паперу [1].

## 2. Загальний опис алгоритму

### 2.1. Алгоритм пошуку та локалізації штрих-кодових об'єктів

Алгоритм пошуку та локалізації штрих-кодів містить частину, яка визначає ділянку зображення, що містить штрихову символіку. Після того, як був знайдений штрих-код, потрібно його локалізувати, тобто знайти межі області, яку він займає. Потім потрібно знайти кут повороту об'єкта з штрих-кодом відносно лінії сканування графічного образу листа. Для пошуку та визначення кута повороту будемо користуватися такими підходами:

Метод пошуку:

- сегментація графічного образу аркуша формату А4 з наклеєними мітками на окремі графічні елементи з можливою наявністю на цих елементах бар-кодів з функцією фільтрації випадкових величин;
- Визначення граничних областей та графічного центру елементу з можливою наявністю бар-коду, локалізації цього елементу.

Метод визначення кута повороту:

- 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 елементу з бар-кодом;
- Визначення кута нахилу елементу відносно лінії сканування графічного образу аркуша.

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

Після того, як штрихові символи знайдені, локалізовані та знайдені їхні кути нахилу, потрібно сформувати вектор яскравості уздовж прямої, що перетинає всі штрихи коду. Це робить алгоритм *виділення структурних елементів*, у якому здійснюється переведення нечітких значень ширини елементів символіки (штрихів і пропусків) в числа, кратні модулю (елементу мінімальної ширини) штрихового коду. На виході алгоритму отримаємо цілочисловий вектор значень ширини елементів. Для цього будуть використані такі підходи:

Метод зчитування бар-коду:

- Сканування в області геометричного центру елементу з врахуванням кута нахилу та пошук мінімальної за розмірами темної та світлої областей;
- Присвоєння значення, що дорівнює кратному мінімальному розміру для всіх світлих та темних областей;
- Зміщення від геометричного центру на певну величину для нового сканування у випадку виявлення помилок у бар-коді.

Всі вищеперелічені операції будуть реалізовані відповідними методами у класі MRecognize.

## 2.3. Метод декодування штрих-кодів

Після того, як отримано вектор із цілочисловими значеннями ширини елементів штрихового коду – все, що залишається зробити – це декодувати його, тобто перетворити штрих-код на його цифровий еквівалент відповідно до стандарту.

Методи, які реалізуватимуть алгоритми декодування для стандарту Code 128, міститимуться в класі MBarcode.

При розробці декодуючих алгоритмів потрібно врахувати такі параметри, як надійність та швидкодія.

Метод декодування:

- 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.

Для досягнення високої швидкодії при аналізі зображень необхідно використовувати принципи розпаралелювання процесів [5]. У розроблюваному алгоритмі локалізації штрих-кодів найчастіше використовуваною операцією є фільтрація за допомогою матриць розміром 2x2 і 3x3. Ця операція використовує дані тільки з локальної і наперед відомої області обробки, вимагає значно меншого об'єму пам'яті, ніж ті операції, початкові дані для яких розкидані по всьому зображенню або розташування яких істотно залежить від результатів попередніх обчислень. Ця частина алгоритму має бути також адаптована для виконання паралельно декількох процесів фільтрації.

Алгоритм виділення структурних елементів штрих-коду повинен бути розроблений, виходячи із того, що розмір і положення вхідних даних є невідомими. У алгоритмі також застосовуються ітераційні методи для визначення розташування штрихів і пропусків. Отже, обсяг використовуваної пам'яті і режим роботи алгоритму змінюються залежно від вхідних даних. Цей алгоритм повністю повинен реалізуватися програмно.

Алгоритм декодування штрих-коду – алгоритм верхнього рівня. Він отримує на вхід значно менший об'єм інформації, має велику кількість варіантів роботи залежно від типу декодованого коду. Тому він може бути реалізований тільки програмно.

Також варто передбачити і можливість нерозпізнавання штрихової символіки. У такому випадку оператор повинен мати можливість ввести вручну нерозпізнану інформацію за допомогою ручного сканера або з клавіатури.

### 3. Алгоритм пошуку та локалізації штрих-кодових об'єктів

Якщо взяти до уваги, що штрих-код, як і текст, має досить різке, таке, що багато разів змінюється на невеликих площах, значення яскравості, а також те, що на відміну від тексту ця зміна має певну просторову монотонність уздовж штрихів, то локально аналізуючи властивості текстури зображення, можна розрізнити штрих-код за однонапрямленістю градієнтів яскравості на деякій площі. В алгоритмі використовувалися прості локальні диференціюючі фільтри. Здійснювалася двомірна фільтрація (вираз 2) початкового монохромного зображення з наведеними у таблиці фільтруючими матрицями.

$$c(n_1, n_2) = \sum_{k_1} \sum_{k_2} a(k_1, k_2) \cdot b(n_1 - k_1, n_2 - k_2), \quad (2)$$

де  $a$  – початкове зображення;  $b$  – фільтруюча матриця;  $k_1, k_2$  – індекси, що проходять по всіх елементах фільтруючої матриці.

#### Фільтруючі матриці

1	2	3	4
Горизонтально диференціююча матриця	Вертикально диференціююча матриця	Діагонально (45) диференціююча матриця	Діагонально (135) диференціююча матриця
$\begin{matrix} -1/4 & -1/4 \\ 1/4 & 1/4 \end{matrix}$	$\begin{matrix} 1/4 & -1/4 \\ 1/4 & -1/4 \end{matrix}$	$\begin{matrix} 0 & -1/4 & 0 \\ 1/4 & 0 & -1/4 \\ 0 & 1/4 & 0 \end{matrix}$	$\begin{matrix} 0 & 1/4 & 0 \\ 1/4 & 0 & -1/4 \\ 0 & -1/4 & 0 \end{matrix}$

Дві пари фільтруючих матриць 1 і 2, 3 і 4 є лінійно незалежними і ортогональними. Кожна пара знаходить максимальний відгук значень компонент градієнта яскравості в двох взаємно перпендикулярних напрямках. Лінійний штрих-код має певний напрям штрихів. Отже, усереднене за деякою площею значення модуля компонент градієнта уздовж двох перпендикулярних напрямів, що мають різні кути з лініями штрих-коду, істотно відрізняться. Ця властивість використовується для попереднього оцінювання місць можливого розташування лінійного штрих-коду, а також грубого визначення напрямів штрихів.

Для зменшення об'ємів обчислень, пов'язаних з оцінкою кутів повороту градієнта, вирішено було розбити всі напрями ( $360^\circ$ ) на 4 дискретні напрями (див. рис. 3) на відстані  $45^\circ$  один від одного. Цей факт пояснює необхідність використання двох діагонально диференціюючих матриць.

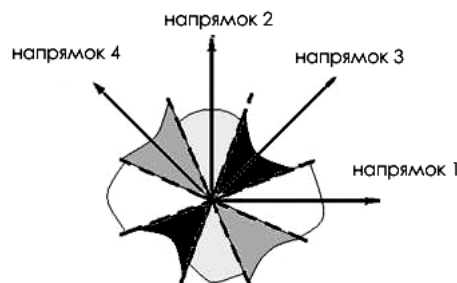


Рис. 3. Дискретність напрямів при аналізі властивостей текстури

Для перевірки достатності виділення лише 4 дискретних напрямів при виявленні лінійного штрих-коду був реалізований такий алгоритм:

- виконується операція фільтрації з фільтрами 1–4;

- для кожного отриманого продукту фільтрації знаходиться сума модулів елементів його складових;
- обчислюється модуль відношень в децибелах результатів для двох пар ортогональних матриць 1, 2 і 3, 4.

На вхід цього алгоритму подавалися зображення постійного розміру 110x110 елементів, що містять відрізок завдовжки 100 елементів. Кут відрізка з горизонтальною віссю зображення змінювався. В результаті була отримана залежність відношення відгуку в двох взаємно перпендикулярних напрямках від кута нахилу відрізка (див. рис. 4).

Водночас були отримані аналітичні вирази для відношень сум модулів відгуків на фільтрації в перпендикулярних напрямках залежно від кута нахилу відрізка, визначених за алгоритмом (рис. 4).

Якщо при виявленні лінійного штрих-коду опиратися на максимальне значення з цих двох відношень для кожного кута нахилу (рис. 4), то мінімальне значення перевищуватиме 6 дБ, тобто цю символіку можна виявити за ослабленням значень відгуку в одному з двох взаємно перпендикулярних напрямках фільтрації більш ніж удвічі.

Продукт фільтрації початкового зображення з матрицями, що фільтрують, є контурним малюнком з контурними лініями, перпендикулярними напрямку диференціювання. Цей малюнок є переривчастим і не може дати інформацію про розмір області, займаній штрих-кодом. Виникає необхідність в усереднюванні значень градієнтів за деякою площею, значно меншою за розміри штрих-коду і більшою за розміри його елементів. Крім того, завданням цього усереднювання є низькочастотна фільтрація зображень з метою зменшення просторового розширення картини градієнтів. Зменшення розширення дасть змогу зменшити обчислювальні навантаження при пошуку областей зображення, що нас цікавлять.

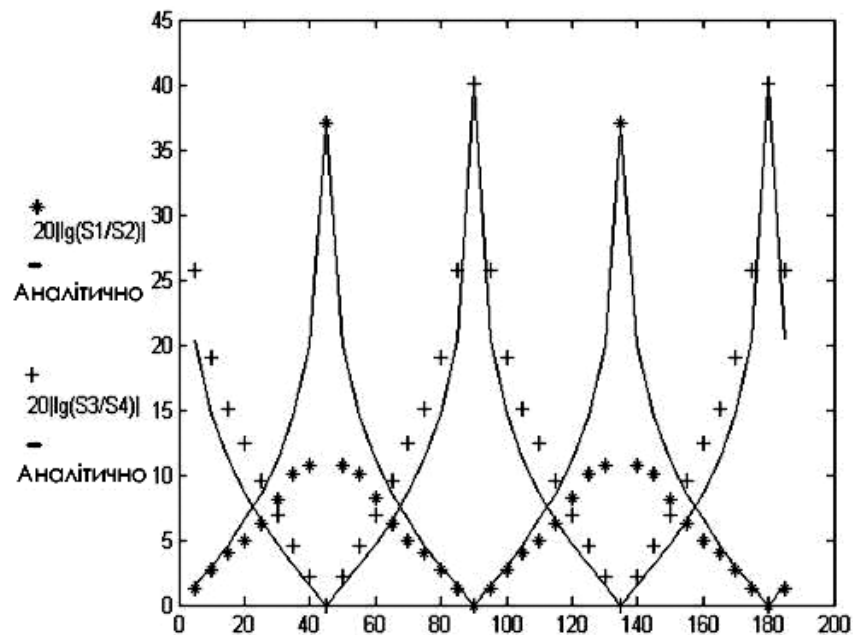


Рис. 4. Графіки відношень сум абсолютних значень відгуків на фільтрацію з фільтрами, в двох парах взаємно перпендикулярних напрямках залежно від кута повороту відрізка прямої лінії у фіксованому квадратному вікні, де  $S1$  – сума модулів відгуків при використанні диференціюючої матриці у горизонтальному напрямі ( $F1$ );  $S2$  – сума модулів відгуків при використанні диференціюючої матриці у вертикальному напрямі ( $G2$ );  $S3$  – сума модулів відгуків при використанні диференціюючої матриці у діагональному напрямі ( $G3$ );  $S4$  – сума модулів відгуків при використанні диференціюючої матриці у діагональному напрямі ( $H4$ )

Ідеальним рішенням для низькочастотної фільтрації є фільтрація двовимірною гауссовою поверхнею (рис. 4), (далі – двовимірним гауссіаном) – вираз (3) та рис. 5. По-перше, ця поверхня має спектральну характеристику тієї самої форми, що і сам гауссіан, тобто що монотонно спадає (див. рис. 6). По-друге, спектральна характеристика однакова на всіх напрямках. По-третє, для знаходження продукту фільтрації немає необхідності у фільтрації початкового малюнку з квадратною матрицею, оскільки цю операцію можна замінити двома одновимірними фільтраціями уздовж рядків і уздовж стовпців (див. рис. 4, 6).

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (3)$$

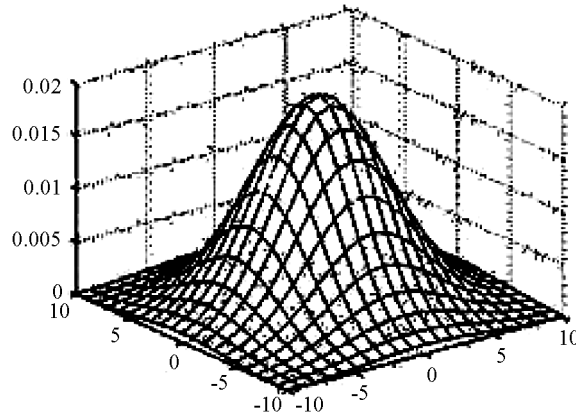


Рис. 5. Двовірний гауссіан

	1	4	7	4	1		
	4	16	26	16	4		
$\frac{1}{273}$	7	26	41	26	7		
	4	16	26	16	4		
	1	4	7	4	1		
<hr style="border: none; border-top: 3px double black;"/>							
	006	061	242	383	242	061	006

Рис. 6. Еквівалентна заміна двовимірного фільтра у формі гауссіана двома одновимірними масивами, призначеними для фільтрації уздовж рядків і уздовж стовпців

Разом з тим, фільтрація за допомогою двовимірного гауссіана поступається за обчислювальними затратами двовимірному одиничному вікню (див. рис. 7), яке так само розглядалося під час розроблення алгоритму. Фільтрації квадратним одиничним вікном – просте підсумовування яскравостей точок за площею усереднювання. У разі фільтрації квадратним одиничним вікном можна спостерігати коливання спектра і неоднорідність отриманої картини за напрямками (див. рис. 8), але це дає незначну похибку у визначенні розташування меж області, займаною штрих-кодом. Тому в описуваному алгоритмі було вирішено використовувати одиничне вікно через можливості використання апаратури з невисокою продуктивністю.

Інформація про мінімальну висоту штрихів (5 мм), проведені експериментальні дослідження, а також попередня оцінка обчислювальних можливостей устаткування, що закладається в проект, привели до висновку, що оптимальне значення зменшення розширення становить за площею в 100 разів, або в 10 разів за лінійними характеристиками.

1/25

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

Рис. 7. Двовимірне одиничне вікно 5×5 елементів

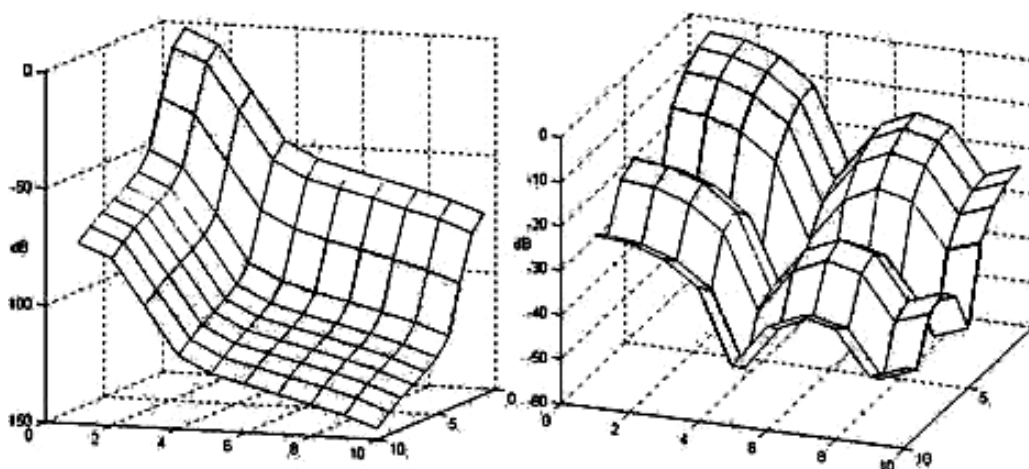


Рис. 8. Спектральна характеристика фільтра Гаусса (ліворуч) і простого квадратного одиничного двовимірного вікна (праворуч)

Нижче наведено приклад роботи алгоритму, на вхід якого було подано зображення, що містить лінійні штрих-коди, орієнтовані під різними кутами (див. рис. 9).

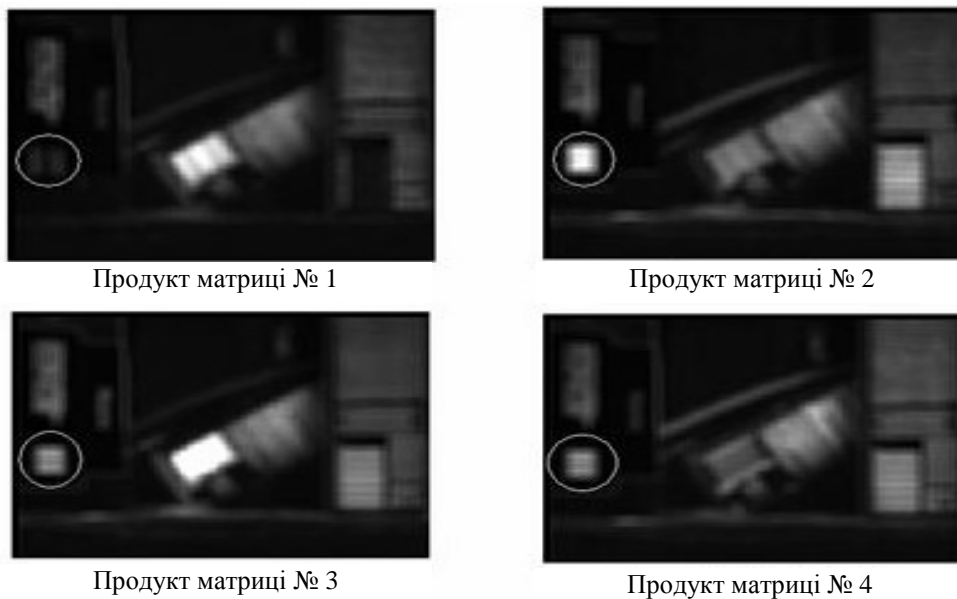


Рис. 9. Початкове зображення сцени

Результати роботи описаного алгоритму можна побачити на рис. 10. Тут показані усереднені за допомогою одиничного вікна продукти фільтрації чотирма наведеними у таблиці матрицями.

Яскравіші області відповідають найбільшому відгуку на фільтрацію з диференціюючими матрицями. Можна зазначити, наприклад, що зображення штрих-кодів із штрихами, паралельними горизонтальному напрямку, мають максимальний відгук при фільтрації вертикально диференціюючою матрицею №2, тоді як при фільтрації горизонтально диференціюючою матрицею №1 їх відгук незначний і практично не відрізняється від 0, відгук на фільтрацію з діагонально диференціюючими матрицями двох типів розрізняється не достатньо (приклад обробки такого штрих-коду обведений кружком на всіх етапах (див. рис. 9, 10, 11)).

Така сама ситуація при фільтрації з діагонально диференціюючими матрицями №3 і №4 спостерігається в місці зображення штрих-коду, нахиленого під кутом, близьким до  $45^\circ$  з горизонтальним напрямом площині картини.



*Рис. 10. Усереднені одиничним квадратним вікном продукти фільтрації початкового монохроматизованого зображення чотирма видами диференціюючих матриць*

Той факт, що при фільтрації з матрицями двох ортогональних напрямів область зображення, що містить лінійний штрих-код, матиме відгук, що розрізняється більше ніж удвічі, хоч би для однієї пари ортогональних матриць, дає змогу визначати області, що містять штрих-код, простим відніманням пар зображень. Алгоритм є таким.

Знаходиться поелементна різниця продуктів фільтрації і усереднювання матриць №1 і №2, а також матриць №3 і №4. Результат проведеної операції зображений на рис. 11. Віднімання двох зображень сцен допомагає позбутися таких схожих за контрастністю на штрих-код об'єктів, як текст. Це вимагає кількості операцій, що дорівнює кількості точок зображення.

Можна зазначити, що після проведення операції поелементної різниці, окрім областей, займаних штрих-кодами, значною яскравістю володіють елементи, що мають певну просторову спрямованість. Такими елементами можуть бути, наприклад, окремо розташовані контурні лінії об'єктів. Усереднювання їх відгуку за площею має малі порівняно з результатами штрих-кодів значення, отже, їх можна позбутися, порівнюючи кожне значення з деяким порогом [6]. Це також вимагає кількості операцій, що дорівнює кількості точок сцени.

Опишемо процедуру вибору порогу. Введемо для розпізнаваних штрих-кодів такі параметри:  
**Strength**="Средня кількість перепадів яскравості на кожні 10 точок";  
**Brigness\_Difference**="Середнє значення різниць яскравості між пропусками і штрихами".



Усереднена яскравість області розміром 10x10 точок після операції фільтрації може бути розрахована:

$$\text{Window\_Brightness} = (2 * (1/4) * \text{Strength} * \text{Brightness\_Difference} * 10) / (10 * 10)$$

Необхідно врахувати, що в перпендикулярному напрямку з максимальним відгуком це саме значення буде як мінімум удвічі меншим – 6 дБ (див. рис. 4)), отже, обмеження порогу згори може бути задане:

$$\text{Max\_Thresh} = \text{Window\_Brightness} - 0.5 * \text{Window\_Brightness} \quad (4)$$

У цьому прикладі зображення штрих-кодів мають такі характеристики:  $\text{Strength} \approx 2$ ;  $\text{Brightness\_Difference} > 0.5$ ; підставивши ці значення у вираз 4, отримаємо, що максимальний поріг дорівнюватиме 0.025.

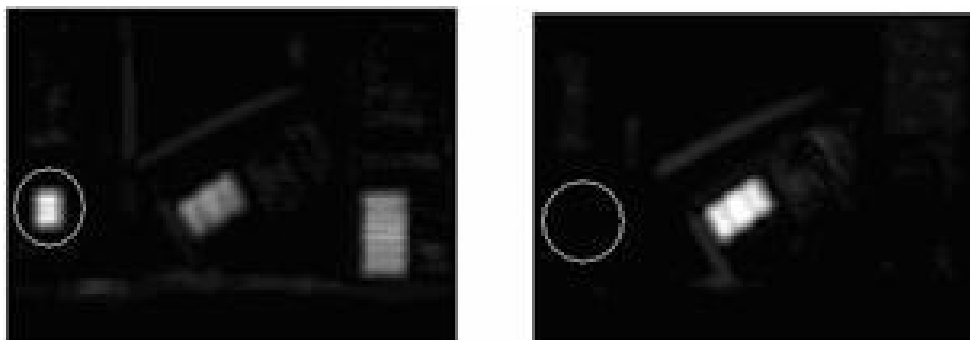


Рис. 11. Абсолютні значення поелементної різниці між усередненими продуктами матриць № 1, № 2 ліворуч і № 3, № 4 праворуч

Із гістограми розподілу яскравостей на малюнках абсолютних значень поелементної різниці, що зображають результат, можна помітити, що більшість значень яскравостей точок лежать нижче за поріг 0.02 (див. рис. 12) Для продовження прикладу порівняємо значення яскравостей на рис. 11 з порогом, що дорівнює 0.02.

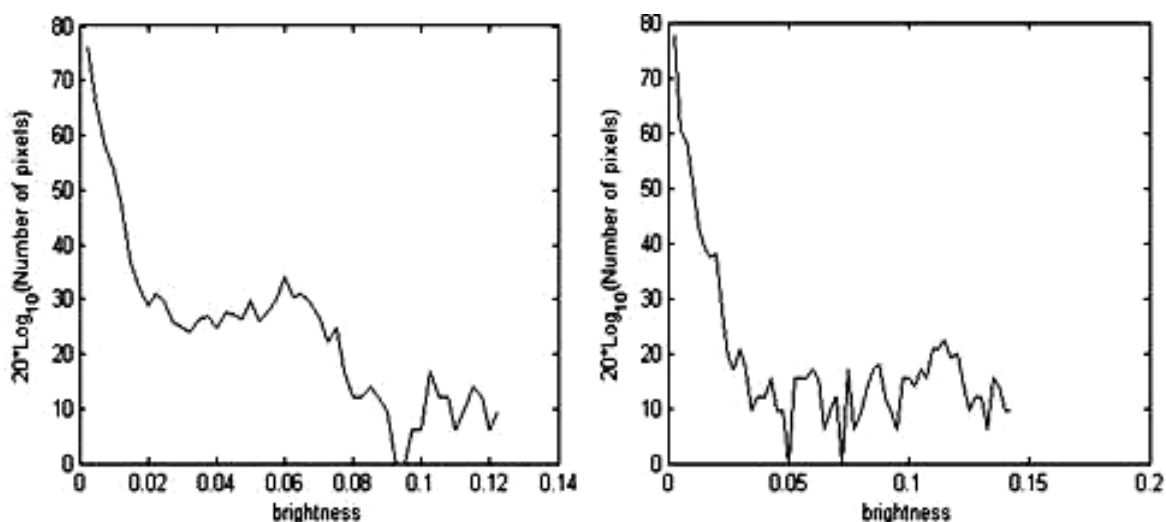


Рис. 12. Гістограма розподілу яскравостей на зображеннях абсолютних значень поелементної різниці між усередненими продуктами матриць № 1, № 2 ліворуч і № 3, № 4 праворуч

Застосування операції порівняння з порогом дасть одиниці у місцях здогадуваного розташування штрих-коду і нулі в усіх інших (див. рис. 13).

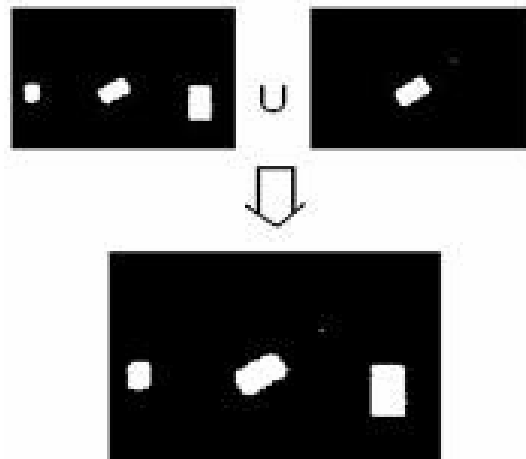


Рис. 13. Застосування операції поріг і об'єднання результатів для двох пар матриць

Маючи таку бінарну картину, для зменшення обчислювальних навантажень можна визначити зони зображення, з якими працюватимуть подальші алгоритми. Для цього необхідно полічити отримані об'єкти, що здійснюється методом нумерації 8-зв'язних об'єктів, описаним в [6]. На виході маємо малюнок, значення точок якого або дорівнюють 0 – фон, або номеру об'єкта. Навколо кожного об'єкта виділяється прямокутна область, яка містить всі точки, що належать конкретному штрих-коду. Якщо площа цієї області менша за деяку константу, її не розглядають.

### Висновки

Під час розроблення методу враховувались такі параметри, як надійність, швидкодія, можливість застосування принципів розпаралелення процедур, економічність апаратного рішення.

Описаний метод використовує таку властивість штрих-кодів, як просторова монотонність в одному напрямі, тому він може бути з успіхом застосований для локалізації багаторядкових і модульованих за висотою символік, описаних у [7].

У розробленому алгоритмі локалізації штрих-кодів основною операцією є фільтрація за допомогою матриць розміром 2x2 і 3x3. Ця операція використовує дані тільки з локальної і наперед відомої області обробки, що вимагає значно меншого обсягу пам'яті. Враховуючи також об'єм вхідних даних 1600x1200x5=9.6МБ/сек, ця частина алгоритму є адаптована для розпаралелення операцій фільтрації.

Розроблення алгоритму виділення структурних елементів та алгоритм декодування штрих-коду будуть детально описані в продовженні.

1. Демида Б., Горон М.. Система автоматизації обліку на швейному виробництві // Вісник Нац. ун-ту "Львівська політехніка", 2008. 2. <http://www.consumerinfo.org.ua>. Штрих-код. 3. Краснобаев А.А. Алгоритмы распознавания штриховых кодов. Дис... канд. техн. наук. – М.: ИПМ им. М.В. Келдыша РАН, 2004. 4. Roger C. Palmer, *The bar code book*. Helmers Publishing, Inc. Formerly North American Technology, Inc., 1995. 5. Кинсун Фу и др. СБИС для распознавания образов и обработки изображений. – М.: Мир, 1988. – 247 с. 6. E.R. Davies, *Machine Vision*.