

УДК 004.315.5 : 315.8

Л.І. ТИМЧЕНКО¹, М.С. ПЕТРОВСЬКИЙ¹, Н.І. КОКРЯЦЬКА¹, Н.П. БАБЮК²

МЕТОД КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ ПЛЯМ ЛАЗЕРНИХ ПУЧКІВ ІЗ ЗАСТОСУВАННЯМ ПАРАЛЕЛЬНО-ІЄРАРХІЧНОЇ МЕРЕЖІ ІЗ ПІДВИЩЕНОЮ ТОЧНІСТЮ

¹*Державний економіко-технологічний університет транспорту,
вул. М. Лукашевича, 19, м. Київ, Україна, E-mail: timchen@list.ru*

²*Вінницький національний технічний університет
21021, вул. Хмельницьке шосе, 95, м. Вінниця, Україна, E-mail: nbabyuk@gmail.com*

Анотація. У статті розглядається метод класифікації зображень плям лазерних пучків і його реалізація. Обговорюється необхідність фільтрації зашумлених зображень адаптивними методами, такими як паралельно-ієрархічні мережі. Представлена реалізація такої мережі на базі ПЛІС.

Аннотация. В статье рассматривается метод классификации пятен лазерных пучков и его реализация. Обсуждается необходимость фильтрации зашумленных изображений адаптивными методами, такими как параллельно-иерархические сети. Представленная реализация такой сети на базе ПЛИС.

Abstract. This article discusses a method of classifying of laser beams spots and its implementation. In article explained necessity of filtering noisy images using the adaptive methods such as parallel-hierarchical network. Represented the implementation of such networks on FPGA.

Ключові слова:

Паралельно-ієрархічні мережі, ПЛІС, лазерні пучки, фільтрація, класифікація, передобробка

ВСТУП

Одним із напрямів сучасних систем комунікації, що швидко розвиваються є супутникові і космічні лазерні лінії зв'язку [1,2].

Увага до подібних систем викликана на сам перед двома причинами:

1. Енергетична ефективність – лазерним системам передачі необхідно значно менше енергії ніж класичним радіосистемам, для передачі на ту ж відстань, через мале розходження променя.

2. Швидкість – при тій же потужності лазерні системи дозволяють передавати значно більше інформації. Що є необхідним у подальших дослідженнях сонячної системи.

Проте мале розходження променя є й однією з основних технічних складностей у побудові подібних систем. Оскільки у приймаючий пристрій необхідно влучити променем з відстані десятки-сотні тисяч кілометрів, і при цьому, передавач постійно рухається. А отже приймач має відслідковувати його положення.

Для таких пристроїв необхідно знаходити координати супутника з високою точністю. Проте проходячи крізь атмосферу лазерне випромінювання зазнає значних перешкод.

Якщо ж енергетичне ослаблення лазерного променя при розповсюдженні його в атмосфері призводить до обмеження дальності дії лазерних пристроїв комунікації та навігації, то його вплив на структуру пучка і траєкторію його розповсюдження визначає в певному ступені параметри точності таких пристроїв. Світлова хвиля, що розповсюджується в атмосфері, крім енергетичних втрат має флуктуації амплітуди і фази. Ці флуктуації відбуваються завдяки випадково-часовому розподіленню показника заломлення, яке, в свою чергу, обумовлене турбулентними рухами в атмосфері, що тягнуть за собою випадкову зміну температури.

Турбулентність атмосфери викликає викривлення хвильового фронту хвилі, яке призводить до розширення лазерного пучка, перерозподілу енергії всередині пучка і зміни його “центра ваги”, а також до пов’язаних з цими явищами флуктуаціями загальної інтенсивності [3,4].

Звичайно вимірювання координат об’єктів відбувається за допомогою геометричних моментів [5-7], використання яких в задачах вимірювання координат енергетичних центрів лазерних зображень, які деформовані атмосферою, призводить до суттєвої похибки при вимірюванні. Похибка вимірювання координат енергетичних центрів лазерних зображень складає до 10 пікселів. Тому актуальною становиться задача класифікації зображень протяжної лазерної траси, у випадку коли на зображення діють багато деформуючих факторів. При цьому зберігаючи паритет між швидкістю обчислень і енергоефективністю схеми.

АЛГОРИТМ ПРЯМОГО-ПАРАЛЕЛЬНО ІЄРАРХІЧНОГО ПЕРЕТВОРЕННЯ

Зазвичай для корекції енергетичного «центру ваги» використовують добре опрацьовані методи, що використовують фільтрацію та класифікацію зображень, і не вимагають складних обчислень. До таких методів відноситься й розроблений та реалізований на ПЛІС авторами метод перерізів [8]. Проте на більших відстанях необхідна адаптивна фільтрація, для чого можна використати такий клас методів класифікації, як нейронні мережі.

Одним з напрямів інтерпретації роботи нейронних мереж є представлення його у вигляді збуджень, які одночасно поширюються, і у якому дія однієї нейронної мережі збуджує відповідний відгук в іншій. Тоді сприйняття часу і причинно-наслідкових зв’язків, настільки складні для програмування, закладаються в архітектуру таких нейронних систем, і такі мережі *називаються паралельно-ієрархічними* (ПІ)[9]. Розглянемо детальніше алгоритм роботи такої мережі при прямому перетворенні зображення:

Розглянемо мережевий процес прямого паралельно-ієрархічного перетворення[10]. Нехай є S ($S = 1, 2, 3 \dots$) непустих множин елементів, що задають інформацію. Кількість елементів множини назвемо його довжиною (позначення L_μ - довжина множини μ). Кількість різноманітних елементів множини назвемо розмірністю даної множини (позначення R_μ).

Розглянемо математичну модель паралельного розкладання множини $\mu = \{a_i\}, i = \overline{1, n}$. За наступною моделлю:

$$\sum_{i=1}^n a_i = \sum_{j=1}^R \left(n - \sum_{k=0}^{j-1} n_k \right) (a^j - a^{j-1}), \quad (1)$$

де $a_i \neq 0$, R - розмірність даної множини. З однакових елементів сформуємо підмножини, елементи однієї підмножини позначимо через $a^k, k = \overline{1, R}$, n_k - кількість елементів у k -ій підмножині (тобто кратність числа a_k), a^j - довільний елемент множини $\{a^k\}$, обраний на j -му кроці, $j = \overline{1, R}$, $a^0 = 0, n_0 = 0$.

Якщо в моделі (1) здійснити заміну знака \sum на об’єднання \cup , то результатом розкладання множини μ є об’єднання елементів (множина розкладання μ^1).

$$\mu = \bigcup_{i=1}^n a_i = \bigcup_{j=1}^R \left(n - \sum_{k=0}^{j-1} n_k \right) (a^j - a^{j-1}) \quad (2)$$

Перетворення множини μ в множину μ^1 , яка задається моделлю (2), назвемо оператором **перетворення G**, тобто

$$G(\mu) = \mu^1 \quad (3)$$

Якщо для вихідних S масивів застосуємо оператор перетворення G , що задається формулою (2), то для кожного масиву одержимо своє порядкове розкладання:

$$\mu_1^1 = \bigcup_{i=1}^{R_1^1} a_{1i}^1, \mu_2^1 = \bigcup_{i=1}^{R_2^1} a_{2i}^1, \dots, \mu_s^1 = \bigcup_{i=1}^{R_s^1} a_{si}^1, \quad (4)$$

де μ_s^1 - множина під номером S на першому рівні, тоді для k -го рівня множина під номером 1 запишеться μ_1^k , відповідно R_s^1 - кількість елементів у S множині на першому рівні, R_k^1 - кількість елементів у S множині на k -му рівні. Об'єднаємо отримані елементи (4) у матрицю M_1 . Назвемо її матрицею прямого розкладання на першому рівні. Тоді вона буде мати вид:

$$M_1 = \begin{pmatrix} a_{11}^1 & a_{12}^1 & \dots & a_{1R_1^1}^1 X & \dots & X \\ a_{21}^1 & a_{22}^1 & \dots & \dots & \dots & a_{2R_2^1}^1 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{s1}^1 & a_{s2}^1 & \dots & a_{sR_s^1}^1 X & \dots & X \end{pmatrix}, \quad (5)$$

або

$$M_1 = \bigcup_{S=1}^s \left(\bigcup_{i=1}^{R_S^1} a_{Si}^1 \right). \quad (6)$$

Для зручності запису вакантні місця в матрицях будемо заповнювати значком " X ". Не порушуючи спільності для наочності другий рядок у (4) має максимальну довжину. Очевидно, що таку довжину може мати будь-який рядок.

Матрицю M_1 перепишемо, згрупувавши її елементи не за рядками, а за стовпчиками і таким способом сформуємо нову матрицю:

$$M_1^T = \bigcup_{i=1}^{R_s^1} \left(\bigcup_{S=1}^s a_{Si}^1 \right), \quad (7)$$

тобто матрицю M_1 - транспонуємо. Позначимо через **T оператор** транспонування матриці, тоді

$$T(M_1) = M_1^T.$$

Починаючи з другого рівня, утворення матриць $M_2^T, M_3^T, \dots, M_k^T$ відбувається не тільки за допомогою транспонування як у (7). Будемо групувати елементи не за стовпчиками, а за діагоналями і отримані множини назвемо діагональними. Позначимо **оператором S** - оператор зсуву рядка на величину меншу номера даного рядка на одиницю і виключення першого стовпчика матриці M_2 в результат розкладання. Таким чином, $S(M_2) = M_2''$, у загальному випадку $S(M_j) = M_j''$, $j = \overline{1, k}$.

Послідовне застосування трьох операторів G, S, T назвемо **функціоналом Ф**, тобто $\Phi(M) = T [S (G (M))]$.

$$\Phi \left[T \left(G \left(\bigcup_{S=1}^s \mu_S \right) \right) \right] = \bigcup_{i=2}^k a_{11}^i, \quad (8)$$

μ_s - початкові множини ($S = 1, 2, 3, \dots$), a_{11}^i - елементи розкладання початкових множин, одержувані по одному на кожному рівні, починаючи з другого.

Таким чином, мережний метод прямого ПП перетворення полягає в послідовному застосуванні до початкових множин $\bigcup_{S=1}^s \mu_S$ по одному разу операторів перетворення G і транспонування T , а потім

$(k-1)$ раз функціонала Φ . На кожному рівні ПІ перетворення утворюються по одному елементу a_{11}^k розкладання S - вихідних множин, де a_{11}^k - вихідна інформація прямого ПІ перетворення. Цією інформацією є діагональні множини з одного діагонального кратного загального елемента.

Для демонстрації моделі розглянемо числовий приклад прямого ПІ перетворення з використанням, наприклад G перетворення, інформація якої задана у вигляді числових множин μ_1, μ_2, μ_3 :

$$\mu_1 = \begin{pmatrix} 2 \\ 3 \\ 5 \\ 7 \end{pmatrix} \quad \mu_2 = \begin{pmatrix} 1 \\ 4 \\ 9 \\ 10 \end{pmatrix} \quad \mu_3 = \begin{pmatrix} 3 \\ 15 \\ 20 \end{pmatrix}$$

Застосовуючи модель виду (2.13), одержимо:

$$\begin{aligned} G(\mu_1) &= (8 \ 3 \ 4 \ 2) \\ G(\mu_2) &= (4 \ 9 \ 10 \ 1) \\ G(\mu_3) &= (9 \ 24 \ 5) \end{aligned}$$

З отриманих результатів побудуємо матриці:

$$M_1 = \begin{pmatrix} 8 & 3 & 4 & 2 \\ 4 & 9 & 10 & 1 \\ 9 & 24 & 5 & X \end{pmatrix}$$

$$M_2 = \begin{pmatrix} 12 & 8 & 1 & X & X \\ X & 9 & 12 & 15 & X \\ X & X & 12 & 2 & 5 \\ X & X & X & 2 & 1 \end{pmatrix} \Rightarrow a_{11}^1 = 12$$

$$M_3 = \begin{pmatrix} 16 & 1 & X & X & X \\ X & 3 & 22 & X & X \\ X & X & 6 & 13 & X \\ X & X & X & 2 & 4 \end{pmatrix} \Rightarrow a_{11}^2 = 16$$

$$M_4 = \begin{pmatrix} 2 & 2 & X & X \\ X & 12 & 16 & X \\ X & X & 4 & 11 \\ X & X & X & 4 \end{pmatrix} \Rightarrow a_{11}^3 = 2$$

$$M_5 = \begin{pmatrix} 4 & 10 & X & X \\ X & 8 & 12 & X \\ X & X & 8 & 7 \end{pmatrix} \Rightarrow a_{11}^4 = 4$$

$$M_6 = \begin{pmatrix} 16 & 2 & X \\ X & 16 & 4 \\ X & X & 7 \end{pmatrix} \Rightarrow a_{11}^5 = 16$$

$$M_7 = \begin{pmatrix} 4 & 14 & X \\ X & 8 & 3 \end{pmatrix} \Rightarrow a_{11}^6 = 4$$

$$M_8 = \begin{pmatrix} 16 & 6 \\ X & 3 \end{pmatrix} \Rightarrow a_{11}^7 = 16$$

$$M_9 = (6 \ 3) \Rightarrow a_{11}^8 = 6$$

$$M_{10} = (3) \Rightarrow a_{11}^9 = 3$$

Результат прямого III перетворення:

$$\Phi_{i=2}^{10} \left[T \left(G \left(\bigcup_{i=1}^3 \mu_i \right) \right) \right] = (12, 16, 2, 4, 16, 4, 16, 6, 3). \quad \sum_{i=1}^{10} \mu_i = 79.$$

З огляду на обчислювальну складність алгоритму прямого III перетворення з позиції схемотехніки, можна зробити висновок, що найбільш трудомісткою є операція G перетворення, операції ж S і T за своєю суттю є операціями з пам'яттю, а отже потребують менш складних апаратних засобів.

2. РОЗРОБКА МЕТОДУ «ГРУБО-ТОЧНОЇ ОЦІНКИ» ПЛЯМ ЛАЗЕРНИХ ПУЧКІВ З ВИКОРИСТАННЯМ МЕТОДУ ПРЯМОГО-ПАРАЛЕЛЬНО ІЄРАРХІЧНОГО ПЕРЕТВОРЕННЯ

Для вирішення поставленої задачі можливо використати двоетапний алгоритм класифікації плям лазерних пучків.

На першому етапі якого знаходяться координати зображення і проводиться сегментація зображень плям лазерних пучків, наприклад на базі паралельної схеми запропонованого авторами методу перерізів[8].

Зображення лазерних плям мають перепад яскравості, при переході до зображення лазерної плями, що чітко видно на гістограмі рис. 1.

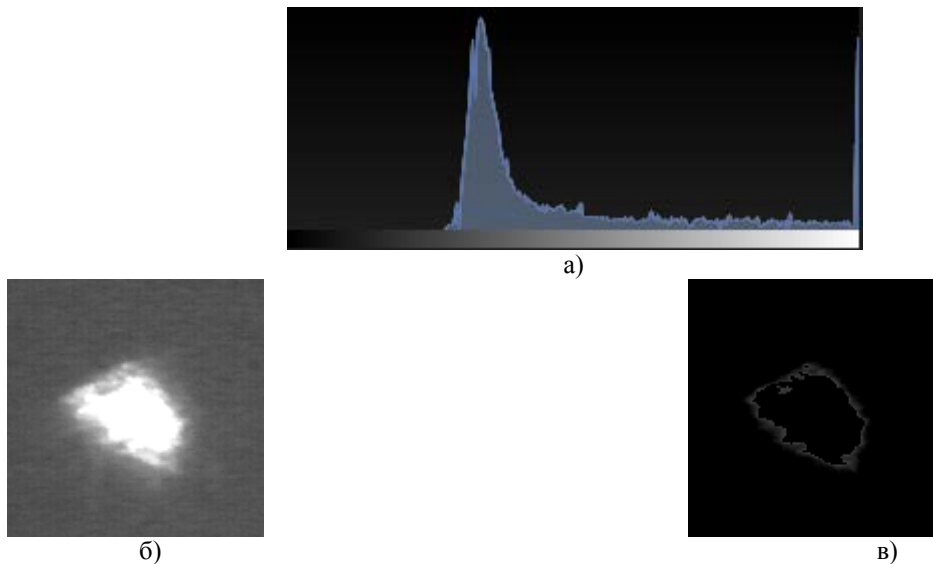


Рис. 1. Гістограма яскравостей (а) плями лазерного пучка (б) і пляма лазерного пучка після сегментації (в)

У лівій половині гістограми знаходяться шуми, які не несуть важливої інформації для III перетворення. А отже сегментація може бути проведена пороговим фільтром з подвійною границею [11], за наступною формулою:

$$\begin{cases} w(f(x, y)) = 0, w(f(x, y)) \leq W_{0,5} \\ w(f(x, y)) = w(f(x, y)) - W_{0,5}, W_{0,5} < w(f(x, y)) < W_{0,75} \\ w(f(x, y)) = 0, w(f(x, y)) \geq W_{0,75} \\ W_{0,75} = W_{\max} * 0,75 \text{ і } W_{0,5} = 0,5 * W_{\max} \end{cases} \quad (9)$$

Де $w(f(x, y))$ - значення яскравості точки, W_{\max} - максимальної яскравості зображення.

На другому ж етапі проводиться кореляційна оцінка хвостових елементів зображення за методом прямого-паралельно ієрархічного перетворення.

Таким чином загальний алгоритм високоточної класифікації плям лазерних пучків виглядає наступним чином:

1. Знаходимо координати зображення, і проводимо сегментацію;
2. Знаходимо хвостові елементи для збереженого зображення за методом прямого III перетворення.

3. Формуємо еталонне зображення – середні значення хвостових елементів перших 20 зображень.
4. Обчислюємо коефіцієнт кореляції між хвостовими елементами еталонного зображення і поточним зображенням.
5. Якщо коефіцієнт кореляції вище 0,9 то зображення «хороше».
6. Повторюємо пункт 4-6 для кожного наступного зображення.

Для створення схеми обробки зображень проводилось моделювання алгоритму на комп'ютері на реальних знімках лазерних трас, з метою виявлення характеристик зображень важливих для створення схеми.

Як наслідок, моделювання прямого III перетворення показало, що максимальна кількість рядків початкової матриці при перетворенні не перевищує 120, а максимальне число не перевищує 2^{10} .

Таким чином, можна задатися такими початковими параметрами для створення схеми III перетворення. Будемо використовувати 16 розрядні слова. Оскільки початкове зображення має розмір 128×128 , то і максимальна кількість рядків для Gперетворення, теж складає 128.

3. РОЗРОБКА СХЕМИ ПРИСТРОЮ КЛАСИФІКАЦІЇ ПЛЯМ ЛАЗЕРНИХ ПУЧКІВ МЕТОДУ «ГРУБО-ТОЧНОЇ» ОЦІНКИ ПЛЯМ ЛАЗЕРНИХ ПУЧКІВ З ВИКОРИСТАННЯМ МЕТОДУ ПРЯМОГО-ПАРАЛЕЛЬНО ІЄРАРХІЧНОГО ПЕРЕТВОРЕННЯ

Відповідно до вимог викладених у попередньому пункті загальна блок схема розробленого пристрою представлена на рис. 2. Моделювання проводилося з використанням програми QuartusIIкомпаніїAltera.

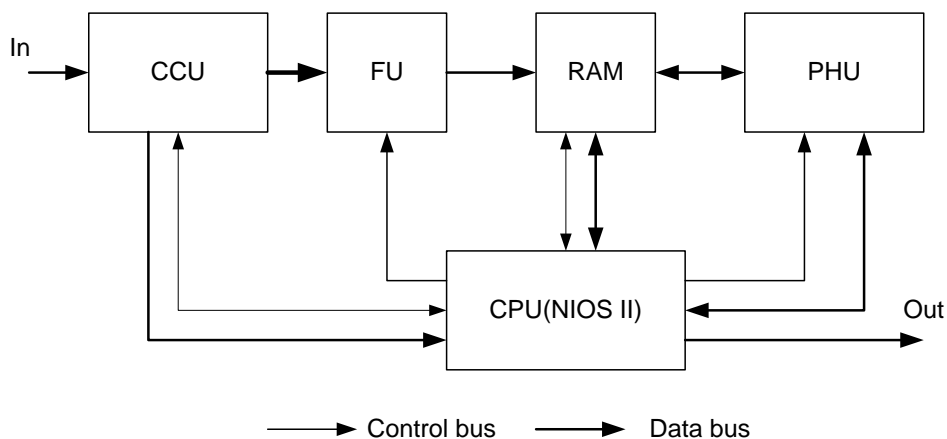


Рис.2. Блок схема пристрою класифікації плям лазерних пучків на базі прямого III перетворення

Пристрій працює наступним чином:

Дані з камери подаються на вхід CCU- coordinatecalculationunit – що знаходить координати енергетичного центру, а також значення $W_{0.5}$ і $W_{0.75}$, що виведені на зовні блоку для подальшої обробки у пороговому фільтрі – FU (filterunit). Дані після фільтра потрапляють до запам'ятовуючого пристрою RAM. І далі потрапляють у блок паралельно-ієрархічного перетворення PHU. На виході цього блоку формуються дані G перетворення.

Керує всім процесом вбудований контролерCPU(типу NIOS II), він же і відповідає за зовнішню комунікацію (Out).

Блок паралельно-ієрархічного перетворення (PHU), що представлений на рис. 3. Опис виводів схеми представлений в таб1.

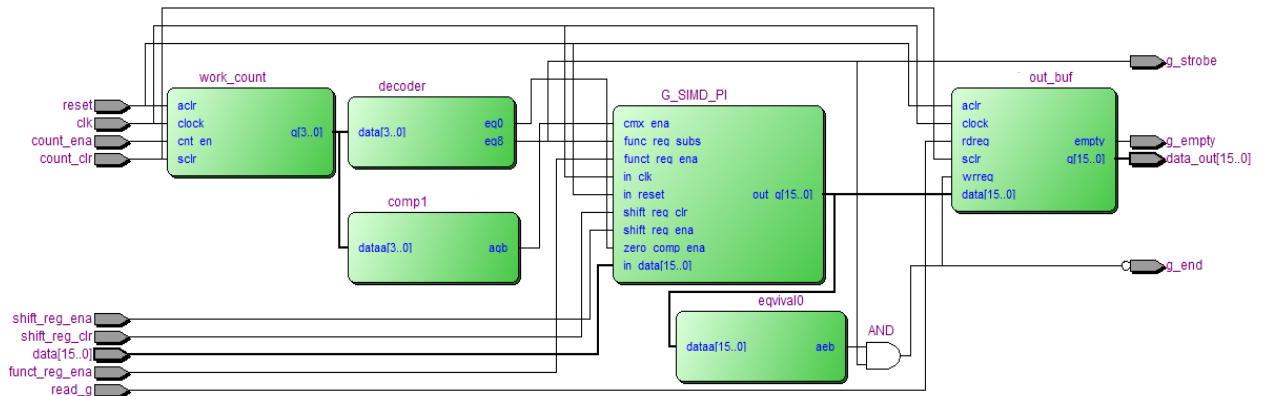


Рис.3. Блок паралельно-ієрархічного перетворення

В цей блок введений регістру зсуву (знаходиться в блоці G_SIMD_PI), що дозволяє робити послідовний ввід даних, а також пристрій керування (кінцевий автомат) на базі лічильника work_count і декодера (decoder). Отже схема працює наступним чином:

На першому етапі, після загального скидання схеми, її приводять до початкового стану поданням «1» на протязі одного такту на вхід shift_reg_clr. Далі подають «1» на вхід shift_reg_ena і через шину даних data завантажують послідовно дані у регістр зсуву.

Таблиця 1.

Виводи блоку паралельно-ієрархічного перетворення

Шина	Тип	Опис
reset	Вхід	Загальне скидання схеми
clk	Вхід	Вхід тактової частоти;
count_ena	Вхід	Вхід дозволу роботи лічильника
count_clr	Вхід	Скидання лічильника
shift_reg_ena	Вхід	Вхід дозволу роботи регістру зсуву
shift_reg_clr	Вхід	Очистка регістру зсуву
funct_reg_ena	Вхід	Строб функціонального регістру.
data [16]	Вхід	шина вводу даних
read_g	Вхід	Строб читання з вихідного буфера
read_g	Вихід	Вихід даних (g функції);
g_strobe	Вихід	Вихід індикації обробки одного члена рядка
g_empty	Вихід	Вихід індикації повністю очищеного буфера
g_end	Вихід	Вихід індикації закінчення g перетворення

Далі на подають «1» на протязі одного такту вхід funct_reg_ena, в результаті чого дані з регістру зсуву поступають на вхід функціонально-запам'ятовуючих елементів рис 4:

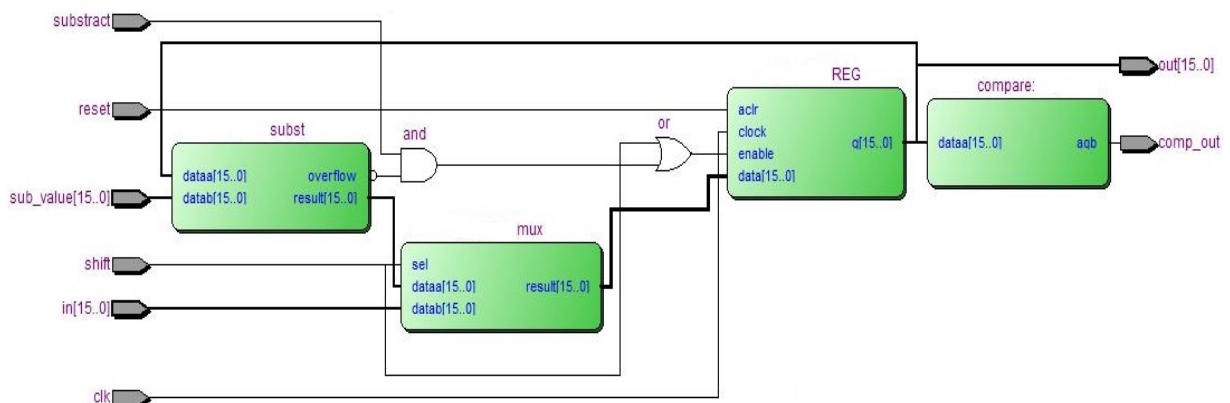


Рис. 4. Функціонально-запам'ятовуючий елемент

Функціонально-запам'ятовуючий елемент складається з наступних блоків: вхідного мультиплектора – mux; регістру зберігання – REG, блока віднімання – subst, та вихідного компаратора compare.

Таблиця 2.

Опис виводів функціонально - запам'ятовуючого елемента

Назва виводу	Напрявлення	Опис
reset	вхід	Початкове скидання
substruct	вхід	Команда проведення віднімання.
shift	Вхід	Команда переносу даних з входу до регістра
clk	Вхід	Тактова частота
in	Вхід	Вхіднашинаданих
out	Вихід	Вихідна шина даних
Sub_value	Вхід	Число, що віднімають
comp_out	Вихід	Вихід компаратора

Функціонально запам'ятовуючий елемент може працювати у двох режимах: завантаження даних, при цьому на вхід shift подається «1». Або віднімання, коли від числа що знаходиться в регістрі REG віднімається, число, що подається на вхід Sub_value, для чого подають «1» на вхід substruct. При цьому результат операції зберігається у тому ж регістрі REG. На виході ж comp_out компаратора –«1» з'являється, тільки тоді, коли в регістрі REG число більше від «0». Часова діаграма роботи даного елемента представлена на рис.5

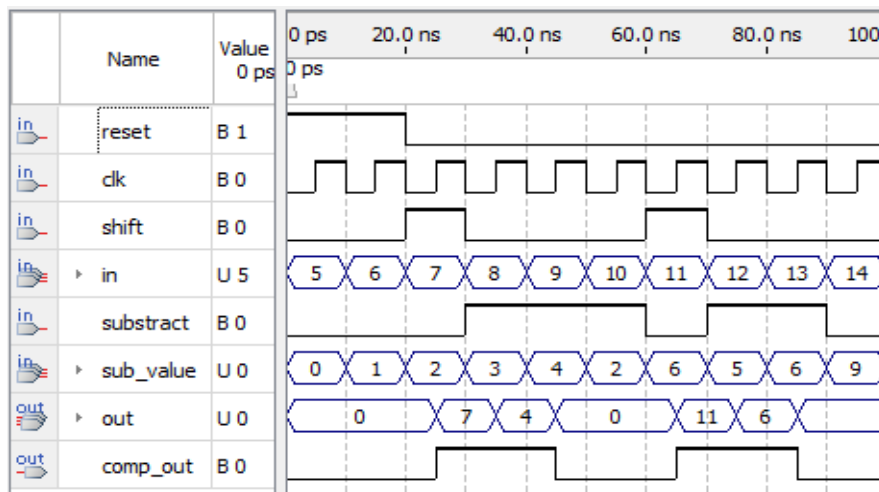


Рис. 5. Діаграма роботи функціонально запам'ятовуючого елемента

На другому етапі подається одиниця на протязі одного такту на вхід count_clr лічильника work_count, що приводить лічильник до початкового стану, а потім його вхід count_ena подають «1», і він починає через декодер (decoder) подавати команди на блок G_SIMD_PI, в результаті чого дані проходять по двійковому дереву[12], елементарні блоки якого представлені на рис. 6, а функції входів у табл. 3.

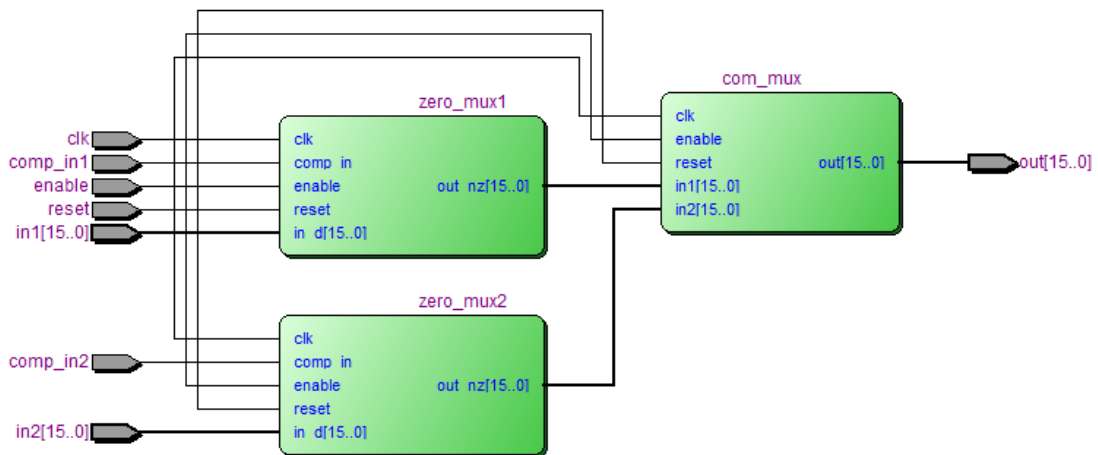


Рис.6. Блоки схеми пошуку мінімального числа

Вхідний мультиплексор zero_mux, слугує для перевірки вхідного значення на «0», і якщо воно рівне нулю (на вході comp_in цього блоку «0»), то на вихід поступає значення Max (максимальне значення для даної розрядності).

Блок com_mux – компаратор-мультиплексор- на своєму виході видає менше з двох чисел, які є на його входах. Варто відзначити, що виходи обох блоків регістрові, а отже робота кожного блоку триває один такт, що представлено на діаграмі роботи схеми на рис. 7.

Таблиця 3.

Опис виводів блоків схеми пошуку мінімального числа

Назва виводу	Напрявлення	Опис
reset	Вхід	Початкове скидання
enable	Вхід	Команда дозволу роботи.
clk	Вхід	Тактова частота
comp_in1, comp_in2	Вхід	Команда, що вхідне число більше 0
in1, in2	Вхід	Вхідні шини даних
out	Вихід	Вихідна шина даних

Отже для створення двійкового дерева схеми G перетворення за SIMD архітектурою для вхідного рядка з довжиною L, необхідно L функціонально запам'ятовуючих елементів (ФЗЕ) і блоків вхідних мультиплексорів (БВМ), а також компараторів-мультиплексорів (КМ), що визначаються як:

$$n = L - 1 \quad (10)$$

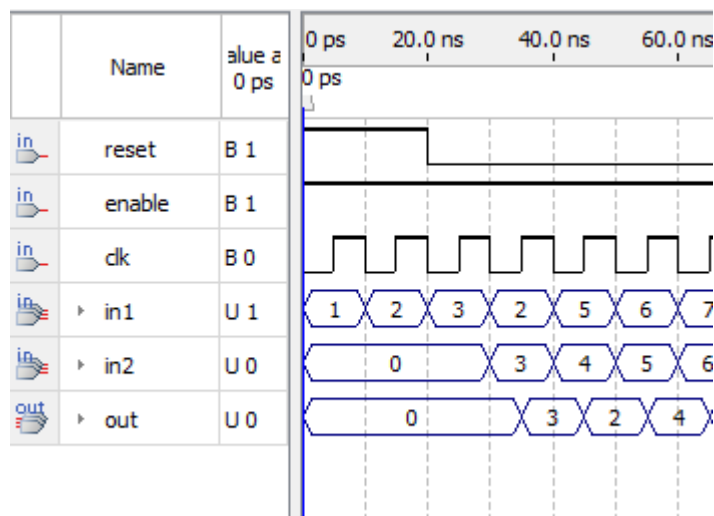


Рис.7. Діаграма роботи схеми двійкового пошуку

В процесі проходження по двійковому дереву знаходиться мінімальне число, і на 8 такті це число помножене на кількість не нульових елементів рядка з'являється на виході блока G_SIMD_PI – тобто перший член G перетворення. Одночасно з цим на виході g_strobe з'являється «1», сигналізуючи про те, що знайдено член G перетворення, і дозволяє контролеру забрати його з буфера не чекаючи закінчення всього перетворення. На 9 ж такті це число заноситься в буфер out_buf.

Далі лічильник починає рахувати спочатку, і через кожні 9 тактів у буфер заноситься новий член G перетворення, так триває доти, доки вхідний рядок не стане рівний нулю, і на виході g_end з'являється одиниця сигналізуючи про те, що перетворення закінчене.

На третьому етапі, контролер подає «1» на вхід read_g, і може зчитувати дані з буфера на виході read_g, по закінченню читання, на виході g_empty з'являється «1».

Очевидно, що кожного наступного рядка даних, робота всіх етапів незалежна, а отже не чекаючи закінчення першого Gперетворення, можливо завантажити у регістр зсуву дані наступного рядка, і також одночасно з Gперетворенням можливо забирати дані з вихідного буфера.

Таким чином час роботи даної схеми у тактах(для найгіршого випадку) становить:

$$T = L(\log_2 L + 2) \quad (11)$$

Моделювання на базі чіпаCycloneIV (EP4CE22F17C6) показало наступні дані необхідні для РНІ:

Total logic elements 11,678 / 22,320 (52 %)
 Totalmemorybits 2,048 / 608,256 (< 1 %)
 EmbeddedMultiplier 9-bit elements 2 / 132 (2 %)
 Fmax-124.32 MHz

Швидкодія ж повно схеми при опрацюванні реальних зображень плям лазерних пучків складала (при основній робочій частоті 100 МГц) складала:

Середня – 7мс/кадр

Максимальний час обробки кадру – 20мс/кадр.

Середня похибка обчислення координат -0.47.

Отже розроблена схема дозволяє проводити і одночасне послідовне завантаження і читання даних.

- Операція G перетворення відбувається самостійно, і не вимагає постійної участі контролера, що розвантажує його для інших операцій.

- Розроблена схема задовольняє початковим вимогам і має запас як по розрядності, так і можливості її нарощування.

4. МОДЕЛЮВАННЯ МЕТОДУ «ГРУБО-ТОЧНОЇ ОЦІНКИ» ПЛЯМ ЛАЗЕРНИХ ПУЧКІВ З ВИКОРИСТАННЯМ ПРЯМОГО-ПАРАЛЕЛЬНО ІЄРАРХІЧНОГО ПЕРЕТВОРЕННЯ

З метою перевірки розроблених алгоритмів та моделей було створено макет пристрою обробки плям лазерних пучків, що представлений на рис. 8.

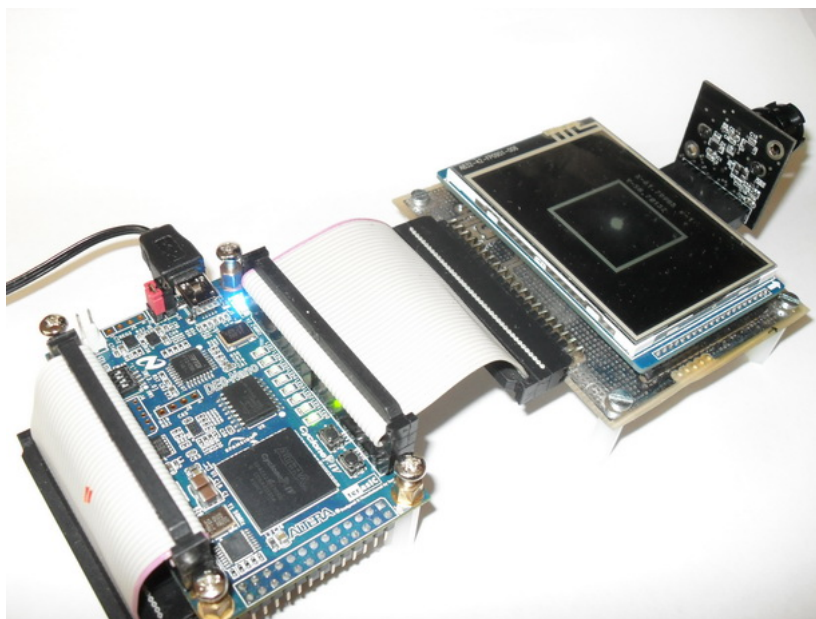


Рис. 8. Макет пристрою обробки плям зелених пучків

Для створення макету були використані наступні наступні компоненти:

- плата розробника DE0-Nano на базі ПЛІС типу AlteraCyclone® IV EP4CE22F17C6N;
- CMOS камера 640*480 на базі чіпу OV7660;
- дисплей TFT 320*240 точок на базі контролера SPFD5408;
- лазерний діод з довжиною хвилі 532 нм (зеленого кольору).

Оскільки ПЛІС має значно менше енергоспоживання у порівнянні з звичайним комп'ютером, то було проведено декілька тестів по споживанню потужності у порівнянні з персональним комп'ютером (на базі процесору AMD Athlon X4 640). Результати порівняння представлені у таблиці 4.

Таблиця 4.

Порівняльна характеристика витрат потужності при обробці плям лазерних пучків з допомогою паралельно-ієрархічних мереж

Платформа	Середня швидкість обчислення, мс	Середня потужність, Вт	Продуктивність, кадрів/вт
Персональний комп'ютер(AMD Athlon X4 640)	45	36	0,62
EP4CE22F17C6N	7	2,5	57

Таким чином з фізичного моделювання (табл.4) можна зробити висновок, що центральний процесор звичайного персонального комп'ютера не може бути використаний для обробки у реальному часі, оскільки середній час обробки більш ніж в двічі перевищує час надходження кадрів (для камери з швидкістю 25кадрів/с , відповідно час обробки одного кадру не може перевищувати 40 мс). До того ж процесор має досить велику споживану потужність у порівнянні з ПЛІС (14.4 раз).

Для систем передачі даних за допомогою лазерного променя важливо знати не тільки точне місце знаходження передавача у даний момент часу, а й вміти прогнозувати його положення у наступний момент часу.

З метою перевірки якості класифікації розробленого алгоритму було проведено моделювання по обробці отриманих координат деякими методами прогнозування на реальних зображеннях лазень трас, що були надані ЗАТ "КІА Системи"(м. Москва, Росія).

У табл.5 представлені дані про середню помилку прогнозування деяких методів прогнозування для попередньої обробки для розроблених методів і методу контурної стрічки[18] (δхср - середня відносна похибка методу, розраховувалася між дійсним і прогнозованим значенням). Дані алгоритми були промодельовані в програмі STATISTICA 6.0. Як видно з таблиці для одних і тих же методів, «грубої-точної» оцінки координат плям лазерних пучків з використанням методу прямого-паралельно ієрархічного перетворення на 15-20% краще, ніж окремо метод перерізів.

Таблиця 5.

Порівняльний аналіз деяких методів прогнозування для різних видів попередньої обробки даних

Метод прогнозування			
Назва методу	Метод перерізів δхср, %	Метод «контурно стрічки», δхср, %	Метод «грубої-точної» оцінки, δхср, %
Числові методи[13]			
1) Експоненціальне згладжування, адитивна модель	0,86	0,87	0,85
2) Експоненціальне згладжування, з демперуванням	0,68	0,69	0,66
3) Авторегресійна модель[14]	1,19	1,43	0,97
Нейронні мережі			
5) Радіально- базисна функція РБФ S5[16]	0,50	0,51	0,43
5) Лінійна мережа S5[17]	0,44	0,46	0,37
6) Багатошаровий перцептрон МР5-2-1 з одним потаємним шаром[19]	0,45%	0,47	0,38

ВИСНОВКИ

В статті розглянуто проблему проходження лазерного випромінювання в атмосфері для атмосферно-оптичних ліній зв'язку. Встановлено, що для забезпечення позиціонування приймально-передавальних пристроїв необхідно класифікувати зображення, з метою виділення не зашумлених зображень.

Для виконання цього завдання з високою точністю запропоновано використовувати метод прямого паралельно-ієрархічного перетворення, що має адаптивні властивості, схожі до нейронних мереж. З метою пришвидшення операцій по обробці плям лазерних пучків було створено комбінований метод «грубої-точної» оцінки лазерних пучків з використанням методу прямого паралельно-ієрархічного перетворення і його реалізацію на ПЛІС. Також створено макет для його перевірки, за допомогою макета встановлено, що:

1. ПЛІС, за рахунок розпаралелювання процесів, має значний вигравш у порівнянні з ПК, так швидкість виконання операцій у 6-10 разів швидше з ПК, а затрачена при цьому енергія майже у 14 разів менша.

2. Розроблений метод попередньої обробки має похибку прогнозування на 15-20% меншу в порівнянні з методом перерізів, та методом контурної стрічки, при використанні одних і тих самих методів прогнозування.

Розроблені у статті методи, також можливо використовувати у лазерній локації та лазерному скануванню.

СПИСОК ЛІТЕРАТУРИ

1. Joshua Buck, Dewayne Washington, NASA Laser Communication System Sets Record with Data Transmission stand from Moon: [Електронний ресурс], режим доступу: World Wide Web: http://www.nasa.gov/mission_pages/ladee
2. Елеутронний портал: [Електронний ресурс], режим доступу: World Wide Web: <http://www.federspace.ru/19038/>
3. A. G. Borovoy, G. Ya. Patrushev, and A. I. Petrov, Laser beam propagation through the turbulent atmosphere with precipitation, Applied Optics, Vol. 27, Issue 17, pp. 3704-3714 (1988)
4. Hugo Weichel, Laser Beam Propagation in the Atmosphere, SPIE, Vo.: TT03, 1990., p-108.
5. Medina-Carnicer, R., Muñoz-Salinas, R., Yeguas-Bolivar, E., Diaz-Mas, L. A novel method to look for the hysteresis threshold for the Canny edge detector, Pattern Recognition, 44 (6), pp. 1201-1211 (2011).
6. Medina-Carnicer, R., Muñoz-Salinas, R., Carmona-Poyato, A., Madrid-Cuevas, F.J. A novel histogram transformation to improve the performance of thresholding methods in edge detection, Pattern Recognition Letters, 32 (5), pp. 676-693 (2011).
7. Fernández-García, N.L., Carmona-Poyato, A., Medina-Carnicer, R., Madrid-Cuevas, F.J. Automatic generation of consensus ground truth for the comparison of edge detection techniques, Image and Vision Computing, 26 (4), pp. 496-511 (2008).
8. M. Petrovski, L. Tymchenko, N. Kokryatskaya, V. Gubernatorov, Y. Kutaev A new sectioning method for classification of optical objects based on PLD. Journal of Computer Vision and Image Processing, Vol. 2, №1, 2012, pp. 33-51.
9. L. I. Timchenko, A multistage parallel-hierarchical network as a model of a neuro-like computation scheme, Cybernetics and Systems Analysis, 36, No. 2, 251-267 (2000).
10. V. P. Kozhemyako, Y. F. Kutaev, L. I. Timchenko, S.V. Chepornyuk, R.R. Hamdi, A.A. Gertsy, I.D. Ivasyuk, The Q-transformation method applying to the facial images normalization, in: Proc. International ICSC/IFAC Symposium on NEURAL COMPUTATION-NC'98, Vienna, 287-291 (1998).
11. William K. Pratt, Introduction to Digital Image Processing, CRC Press, 2013, -708p.
12. Paolo Giudici, Applied data mining: statistical methods for business and industry, John Wiley and Sons, 2003, pp. 111-117.
13. P.J. Brockwell and R.A. Davis, Introduction to Time Series and Forecasting, second edition, Springer-Verlag, New York., 2002,
14. Box George, Jenkins Gwilym, Time series analysis: forecasting and control, rev. 3 Prentice Hall PTR Upper Saddle River, NJ, USA, 1994, 592p.
15. Haykin S., Neural Networks, 2nd ed., New Jersey: Prentice Hall, 1999, -1104p.
16. Dimitri P. Bertsekas, Linear network optimization: algorithms and codes, MIT Press, 1991, pp. 109-122.

17. Robert J. Howlett, L. C. Jain, Radialbasisfunctionnetworks 2: newadvancesindesign, - Haidelberg; NewYork: Phisycs-Verl., 2001.
18. A. Poplavskyy, N. Petrovskiy, L. Timchenko, N. Kokriatskaia, and Y. Kutaev, "Methodofreferencetunnelformationforimprovingforecastresultsofthelaserbeamsspotimagesbehavior" J.Opt. Eng. 50, 117007 (2011), DOI:10.1117/1.3655502
19. Боровиков В.П. Нейронные сети. Методология и технология современного анализа данных / В.П. Боровиков. -М.: Горячая линия-Телеком, 2008. - 392с.

Надійшла до редакції 24.05.2014р.

ТИМЧЕНКО ЛЕОНІД ІВАНОВИЧ – д.т.н., проф, зав. кафедрою телекомунікаційних технологій та автоматики, державний економіко-технологічний університет транспорту, м. Київ, Україна.

ПЕТРОВСЬКИЙ МИКОЛА СЕРГІЙОВИЧ - аспірант кафедри телекомунікаційних технологій та автоматики, державний економіко-технологічний університет транспорту, м. Київ, Україна.

КОКРЯЦЬКА НАТАЛІЯ ІВАНІВНА – к.т.н., доцент кафедри телекомунікаційних технологій та автоматики, державний економіко-технологічний університет транспорту, м. Київ, Україна.

БАБЮК НАТАЛІЯ ПЕТРІВНА – здобувач, Вінницький національний технічний університет, м. Вінниця, Україна.