

УДК 004.932

О.С. БЕЗКРЕВНИЙ, А. В. КОЖЕМ'ЯКО

АВТОМАТИЗОВАНА СИСТЕМА ДЛЯ ОБРОБКИ ЗОБРАЖЕНЬ НА БАЗІ ПЛІС

*Вінницький національний технічний університет
Хмельницьке шосе, 95, м. Вінниця, 21021, Україна
e-mail: alexvntu@gmail.com*

Анотація: Розширення області застосування інтелектуальних систем потребує методів та засобів швидкісного паралельного оброблення значних масивів даних, в тому числі й зображень. Широке застосування автоматизованих систем потребує огляд можливих методів для швидкої роботи з високорозмірними зображеннями, запропоноване рішення має значно прискорити процес паралельного оброблення даних, при апаратній реалізації блоків та підсистем інтелектуальних систем.

Ключові слова: оператор Собеля, ПЛІС, обробка зображень.

Аннотация: Расширение области применения интеллектуальных систем требует методов и средств скоростного параллельного обработки значительных массивов данных, в том числе и изображений. Широкое применение автоматизированных систем требует обзор возможных методов для быстрой работы с высокоразмерными изображениями, предложенное решение должно значительно ускорить процесс параллельного обработки данных, при аппаратной реализации блоков и подсистем интеллектуальных систем.

Ключевые слова: оператор Собеля, ПЛИС, обработка изображений.

Abstract: The expansion of the field of application of intelligent systems requires methods and means of high-speed parallel processing of large data sets, including images. Widespread use of automated systems requires an overview of possible methods for fast work with high-size images, the proposed solution should significantly accelerate the process of parallel data processing, in the hardware implementation of units and subsystems of intelligent systems.

Keywords: Sobel operator, FPGA, image processing.

DOI: 10.31649/1681-7893-2020-39-1-21-26

ВСТУП

Основна інформація міститься не у яскравості окремих областей, а в їх контурах. Завдання виділення контурів полягає в побудові зображення на основі меж об'єктів і контурів однорідних областей.

Будемо називати контуром зображення сукупність його пікселів, в околиці яких спостерігається стрибкоподібна зміна функції яскравості. Так як при цифровій обробці зображення представлено як функція цілочисельних аргументів, то контури представляються лініями шириною, як мінімум, в один піксель. При цьому може виникнути неоднозначність у визначенні лінії контуру. [1] Якщо вихідне зображення, крім областей постійної яскравості, містить ділянки з плавно мінливою яскравістю, то введене визначення контуру залишається справедливим, однак при цьому не гарантується безперервність контурних ліній: розриви контурів будуть спостерігатися в тих місцях, де зміна функції яскравості не є достатньо різкою.

З іншого боку, якщо на зображенні будуть присутні шуми, то будуть виявлені "зайві" контури в точках, які не є межами областей. [2]

ОСНОВНА ЧАСТИНА

Для виділення контурів зображення існує багато математичних методів, будемо використовувати оператор Собеля, який базується на використанні області зображення 3x3, що показано за рис. 1. Він досить схожий на оператор Превітта, а видозміна полягає в використанні вагового коефіцієнта 2 для середніх елементів: [1]

| | | |
|----|----|----|
| -1 | -2 | -1 |
| 0 | 0 | 0 |
| 1 | 2 | 1 |

| | | |
|----|---|---|
| -1 | 0 | 1 |
| -2 | 0 | 2 |
| -1 | 0 | 1 |

Рис. 1 – Маска оператора Собеля

$$G_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3) \quad (1)$$

$$G_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7) \quad (2)$$

Розглянуті вище маски застосовуються для отримання складових градієнта G_x та G_y . Для обчислення величини градієнта ці складові необхідно використовувати спільно:

$$\nabla f \approx |G_x| + |G_y| \quad (3)$$

$$f = \sqrt{G_x^2 + G_y^2} \quad (4)$$

Оскільки мова йде про реалізацію даного методу на ПЛІС, де є можливість паралельної обробки, то доцільним буде розбивати зображення на частини та обробляти їх паралельно і після обробки «склеювати» частини. Виконання даних операцій призначене для збільшення швидкодії, що є доволі актуальним питанням саме для сфери апаратної обробки зображень. [3-4]

Алгоритм для такої задачі буде виглядати наступним чином:

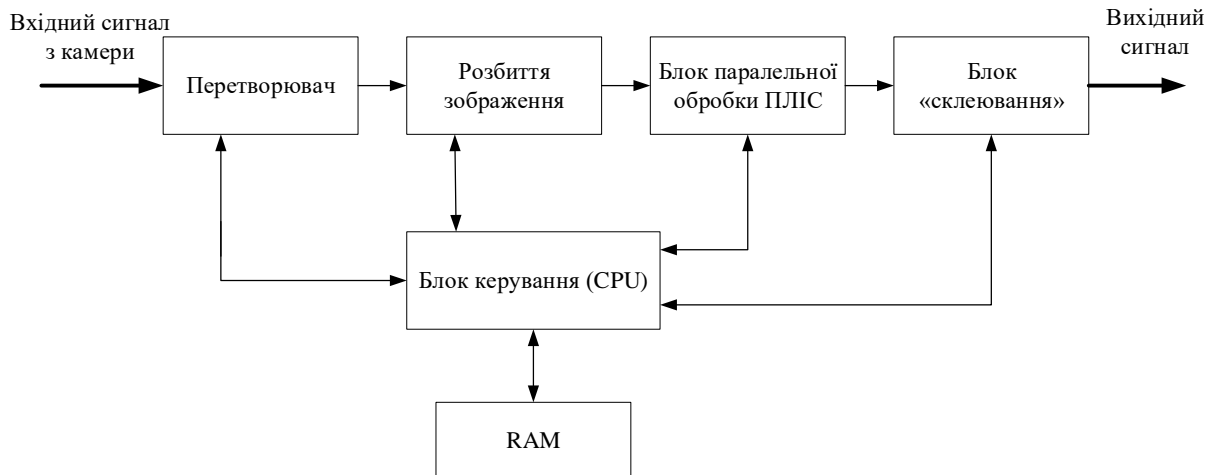


Рис. 2 – Алгоритм передачі даних на ПЛІС та обробки зображення

Дані з камери подаються на вхід перетворювача – що з кольорового RGB зображення перетворює його у градації сірого, для подальшої обробки зображення розбивається на 4 частини – при розмірності 1920*1080 отримуємо 4 зображення з розмірністю 910*540, після чого, зображення подаються на паралельні входи ПЛІС, де паралельно і незалежно одне від одного обробляються, а саме проходять процедуру виділення контурів за методом оператора Собеля. Після завершення даної процедури, частини зображення склеюються в одне з початковою розмірністю в 1920*1080. Всім процесом керує вбудований контролер CPU(типу NIOS II), який також дає команду записувати проміжні етапи в запам'ятовуючий пристрій RAM. Після завершення перетворень і виділення контурів отримане готове зображення передається на пристрій виведення, тощо.

Дане тестове зображення завантажується в систему на базі ПЛІС, і починається його обробка, також варто зауважити, що вхідне зображення в систему може надходити з камери, що дає змогу говорити про автономність роботи даного пристрою.

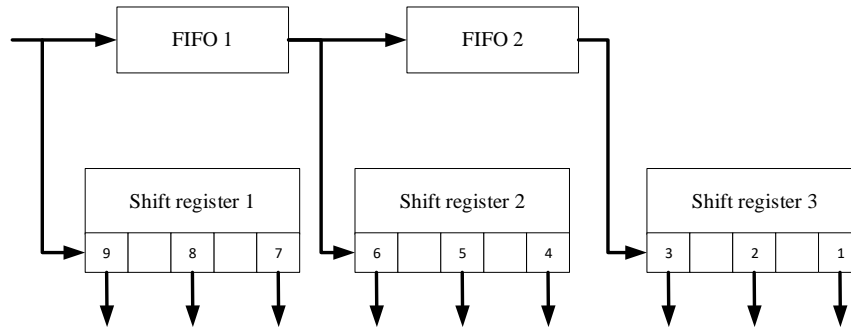


Рис. 3 – Схема ядра оператора Собеля на ПЛІС [5]

Код реалізації ядра оператора Собеля:

```
reg [7:0] a0,b0,c0,a1,b1,c1,a2,b2,c2;
always @(posedge clk or negedge nRst)
if (!nRst) begin
a0 <= 8'd0; b0 <= 8'd0; c0 <= 8'd0;
a1 <= 8'd0; b1 <= 8'd0; c1 <= 8'd0;
a2 <= 8'd0; b2 <= 8'd0; c2 <= 8'd0;
end else begin
a0 <= line1_data;
b0 <= line2_data;
c0 <= line3_data;
//pipeline step 1
a1 <= a0;
b1 <= b0;
c1 <= c0;
//pipeline step 2
a2 <= a1;
b2 <= b1;
c2 <= c1;
end
end
```

Для прикладу буде взято тестове зображення у форматі *.jpg:



Рис. 4 – тестове зображення (вхідне)

Після того як зображення завантажено отримуємо його в кольоровій схемі «градації сірого»:



Рис. 5 – Перетворене зображення з RGB у градації сірого

Наступний крок – «розрізання» зображення в ході якого отримуємо 4 куски зображення:



Рис. 6 – Один з кусків зображення

Після отримання розрізання, кожна з частин паралельно обробляється на ПЛІС, а саме проходить процедуру виділення контурів на основі математичної моделі оператора Собеля, також було розглянуто оператор Щарра – цей метод доволі схожий з методом Собеля, проте його ядро виглядає дещо інакше:

$$\begin{bmatrix} +3 & +10 & +3 \\ 0 & 0 & 0 \\ -3 & -10 & -3 \end{bmatrix} \begin{bmatrix} +3 & 0 & -3 \\ +10 & 0 & -10 \\ +3 & 0 & -3 \end{bmatrix}$$

Рис. 7 – Маска (ядро) оператора Щарра

Проте оператор Щарра часто виділяє хибні об'єкти тому потребує додаткової калібровки та налаштування.

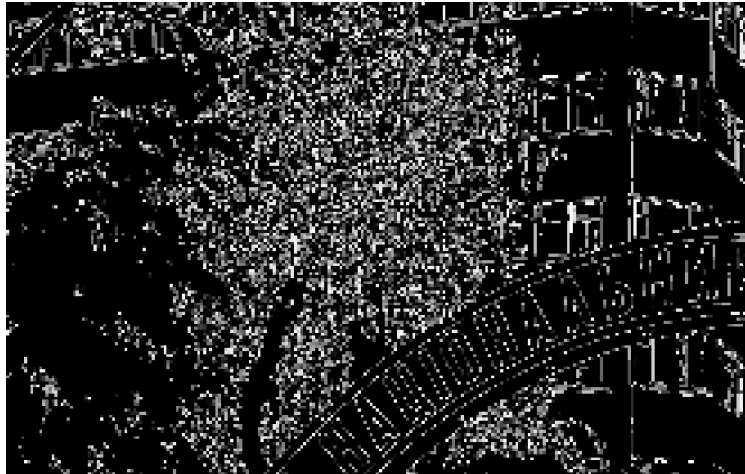


Рис. 8 – Оброблений кусок зображення

Після отримання кусків вони склеюються в ціле зображення і отримуємо оброблене зображення методом Собеля:

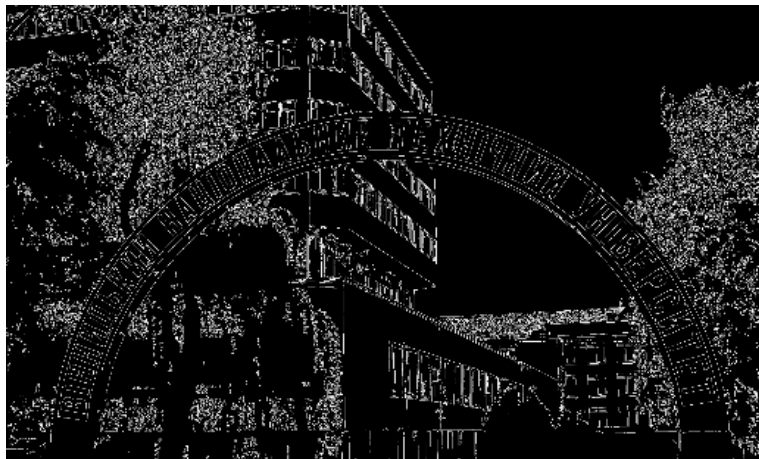


Рис. 9 – Оброблене зображення за методом Собеля в апаратній реалізації на базі ПЛІС

Отримуємо готове зображення оброблене за методом Собеля. Ключовим фактором є те що даний метод реалізовано апаратно з використанням паралелізму, що доступний на ПЛІС`ах і дає можливість значно швидше обробляти зображення великих розмірностей. Для тесту було використано зображення 1920*1080 пікселів.

Код реалізації самого оператора Собеля:

```
module sobel_oper (clk,z0,z1,z2,z3,z4,z5,z6,z7,z8,edge_out);  
  input clk;  
  input [7:0] z0,z1,z2,z3,z4,z5,z6,z7,z8;  
  output [7:0] edge_out;  
  reg signed [10:0] Gx;  
  reg signed [10:0] Gy;  
  reg signed [10:0] abs_Gx;  
  reg signed [10:0] abs_Gy;  
  reg [10:0] sum;  
  always @ (posedge clk) begin  
    //original  
    //Gx<=((z2-z0)+((z5-z3)<<1)+(z8-z6)); //masking in x direction  
    //Gy<=((z0-z6)+((z1-z7)<<1)+(z2-z8)); //masking in y direction  
    // modified  
    Gx <= (z4-z3);  
    Gy <= (z4-z1);  
    abs_Gx <= (Gx[10]?~Gx+1'b1:Gx); //if negative - invert and add to make pos.
```

```
abs_Gy <= (Gy[10]?~Gy+1'b1:Gy);//if negative - invert and add to make pos.  
sum <= abs_Gx+abs_Gy;  
end  
//Apply  
assign edge_out = (sum > 20) ? 8'hff : 8'h00;  
endmodule
```

ВИСНОВОК

Було реалізовано метод Собеля для виділення контурів на ПЛІС з використанням паралелізму, який дозволяє за рахунок розбиття зображення на декілька частин швидше проводити операції над ними розбиваючи великорозмірні зображення.

Апаратна реалізація в свою чергу дозволяє говорити про те що даний процес може бути реалізовано автоматично без керування та втручання з боку людини.

СПИСОК ЛІТЕРАТУРИ

1. Васюра А.С. Методи та засоби нейроподібної обробки даних для систем керування / А.С. Васюра, Т.Б. Мартинюк, Л.М. Куперштейн. – Вінниця: УНІВЕРСУМ-Вінниця, 2008. – 175 с. – ISBN 978-966-641-279-2
2. Мартинюк Т.Б. Адаптивний суматор для систем керування роботом / Т.Б. Мартинюк, А.В. Кожем'яко, Н.В. Фофанова, О.М. Наконечний // Оптико-електронні інформаційно-енергетичні технології. – 2005. – № 2(10). – С. 96-101. – ISSN 1681-7893.
3. Т. Мартинюк, Кожем'яко А., Л. Крупельницький, О. Перебейніс, і О. Безкрєвний, Реалізаційні моделі матричного обчислювача для класифікатора біомедичних даних, ІТКІ, vol 36, № 2, с. 43-51, Груд 2016.
4. Patent for utility model 109748 of Ukraine IPC (2016.01) G06F 12/00, G06F 7/00. A cell of a homogeneous structure / Martyniuk T. B., Kozhemyako A. V., Perebeinis O. M., Bezkrėvnyi O. S. – No U201600094; stated . 01.04.2016; publ . September 12, 2016, Bulletin No 17.

REFERENCES

1. Vasyura AS Methods and means of neuro-like data processing for control systems / A.S. Vasyura, T.B. Martyniuk, L.M. Cooperstein. - Vinnytsia: UNIVERSUM-Vinnytsia, 2008. - 175 p. - ISBN 978-966-641-279-2
2. Martyniuk TB Adaptive adder for robot control systems / T.B. Martyniuk, A.V. Kozhemyako, NV Фофанова, О.М. Nakonechny // Optoelectronic information and energy technologies. - 2005. - № 2 (10). - P. 96-101. - ISSN 1681-7893.
3. T. Martyniuk, Kozhemyako A., L. Krupelnytsky, O. Perebeinis, and O. Bezkrėvny, Implementation models of a matrix calculator for the classifier of biomedical data, ІТКІ, vol 36, № 2, p. 43-51, Dec 2016.
4. Patent for utility model 109748 of Ukraine IPC (2016.01) G06F 12/00, G06F 7/00. A cell of a homogeneous structure / Martyniuk TB, Kozhemyako AV, Perebeinis OM, Bezkrėvnyi OS - No U201600094; stated. 01.04.2016; publ. September 12, 2016, Bulletin No 17.

Безкрєвний Олександр Сергійович – аспірант кафедри ЛОТ, Вінницький національний технічний університет

Кожем'яко Андрій Вікторович – к.т.н., доцент кафедри ЛОТ, Вінницький національний технічний університет