

A possibility for Edge detection using LabVIEW graphical programming environment

Liljana Docheva¹

Abstract – In this paper implementation of different edge detectors using LabVIEW graphical programming environment is given. When a system is implemented with LabVIEW, one can produce highly parallel data flow. This will lead to simplification and to speed increasing of the whole system.

Keywords – Edge detection, LabVIEW.

I. INTRODUCTION

Edge detection is a tool, used in image processing, computer vision and machine vision. Feature detection and feature extraction can't be performed without this tool. From the edges of an image can be extracted very important features and thus the amount of data that will be processed is reduced and the analysis of the image is simplified. Edge detection can be used from higher-level computer vision algorithms (object recognition, motion etc.).

II. REVIEW OF SOME OF THE EDGE DETECTORS

Edges are area of the image with significant changes of the intensity, therefore it mast to decide whether a pixel with particular intensity is edge point or no. There are different detectors that can be applied.

A. The Roberts edge operator

The Roberts edge operator is a gradient based operator.

Approximation of the gradient by the finite differences can be written as (Eq. 1 and Eq. 2) [3]:

$$\frac{\partial f}{\partial x} = f(j-1, k) - f(j, k-1) \quad (1)$$

$$\frac{\partial f}{\partial y} = f(j, k) - f(j-1, k-1) \quad (2)$$

This equations corresponds to the following masks (Eq. 3 and Eq. 4):

$$H_1 = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \quad (3)$$

$$H_2 = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \quad (4)$$

¹ Liljana Docheva is with the Faculty of Telecommunications at Technical University of Sofia, 8 Kl. Ohridski Blvd, Sofia 1000, Bulgaria, E-mail: docheva@tu-sofia.bg.

It can be made the following assumption (Eq. 5 and Eq. 6):

$$G_1(j, k) = \frac{\partial f}{\partial x} \quad (5)$$

$$G_2(j, k) = \frac{\partial f}{\partial y} \quad (6)$$

The equation for the magnitude is.

$$A(j, k) = \sqrt{G_1^2(j, k) + G_2^2(j, k)} \quad (7)$$

Edge orientation can be calculated by equation 8.

$$\alpha = \arctg\left(\frac{G_2}{G_1}\right) + \frac{\pi}{4} \quad (8)$$

B. The Prewitt edge operator

For edge detection with the Prewitt operator the following masks are used (Eq. 9 and Eq. 10)

$$H_1 = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \quad (9)$$

$$H_2 = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad (10)$$

The magnitude can be calculated by Eq. 7. Equation 11 is calculation of the Edge orientation.

$$\alpha = \arctg\left(\frac{G_2}{G_1}\right) \quad (11)$$

A. The Sobel edge operator

For edge detection with the Sobel operator the following masks are used (Eq. 12 and Eq. 13)

$$H_1 = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad (12)$$

$$H_2 = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (13)$$

The magnitude can be calculated by Eq. 7 and the edge orientation by Eq. 11.

A. The Laplasian

ENVIRONMENT

This is edge detector that uses the second derivative (Eq. 14) [2,3]. In this case an edge point can be detected by finding the zero-crossing of the second derivative.

$$\nabla^2 f(x, y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \tag{14}$$

When equation 15 for the Laplasian or equation 7 for the others three edge operators are computed it can be made decision (Eq. 16) whether a pixel belong to the edge or not.

$$A(j, k) = f(j, k) * * H \tag{15}$$

$$A(j, k) \geq \Delta \tag{16}$$

The Laplasian can be implemented using the following mask:

$$H = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \tag{18}$$

The implementation of these four edge detectors using LabVIEW graphical programming environment without any additional tools is given in the paper.

III. IMPLEMENTATION OF EDGE DETECTORS USING LABVIEW GRAPHICAL PROGRAMMING

LabVIEW is a graphical programming environment that uses icons to create applications [1]. One is flexible and easy to solve different kind of problems. Highly parallel data flow that one can produce, make this programming environment suitable for image processing. In this paper is shown a way to implement edge detection using LabVIEW without any additional tools.

Implementation of Gradient based operator like Roberts, Prewitt and Sobel is shown on fig.1. There are differences in the masks (eq. 9 and 10 and eq. 12 and 13 accordingly) and in the determining the direction of the edge (eq. 11).

The first step for the edge detection realization is to read the image file. Then the cluster of image data must to converts into a 2D array. This allows to apply the masks and compute the magnitude A (Eq. 7). Next it can be determined the direction of the edge (α). The last step is to make decision whether a pixel belong to the edge or not.

Figure 2 presents the masks for the Roberts edge operator.

Implementation of the Prewitt and Sobel edge operators is similar. The differences are in the masks (Fig. 3 and Fig. 4 accordingly) and in the determining the direction of the edge (eq. 11).

As it is mention above the Laplasian uses the second derivative. An implementation of this edge operator with LabVIEW has small differences from the one of Gradient

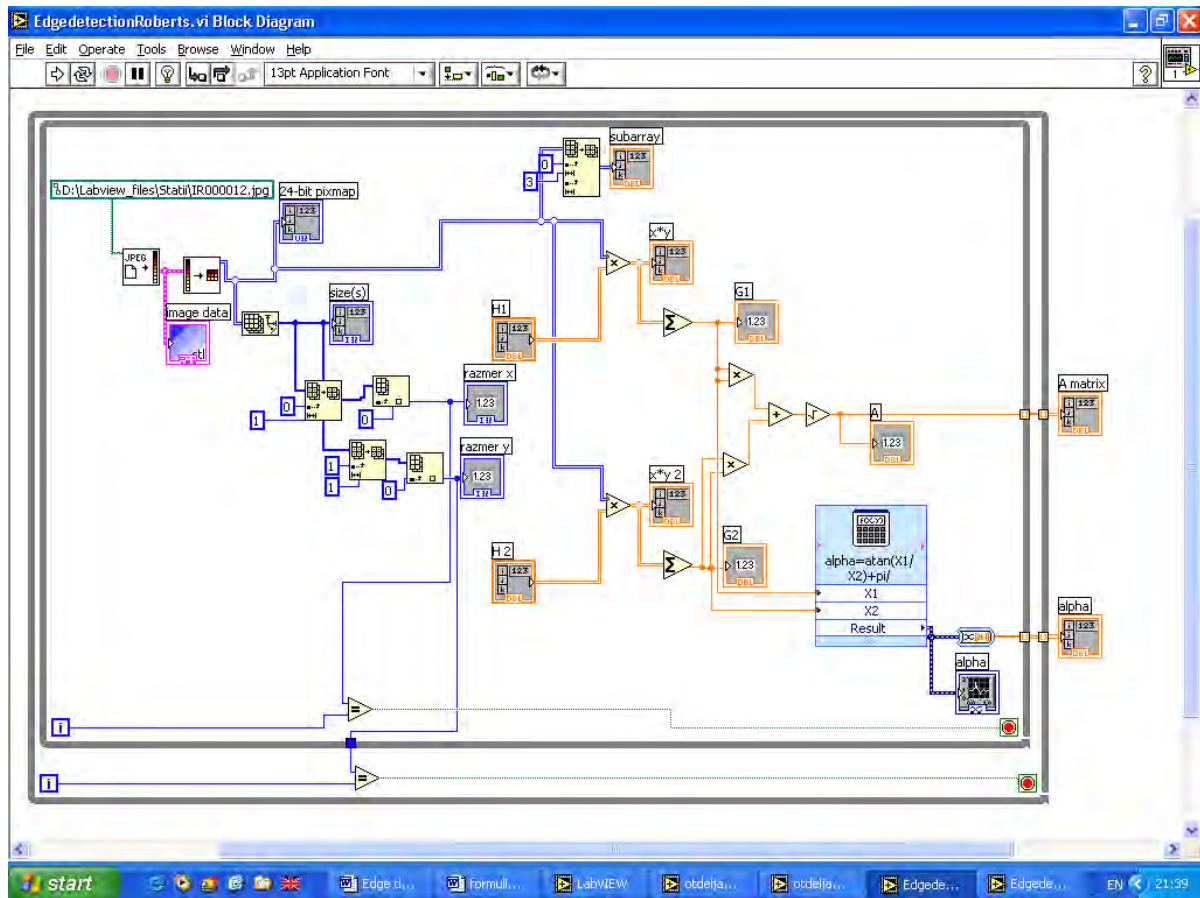


Fig. 1. Implementation of Gradient based operator with LabVIEW.

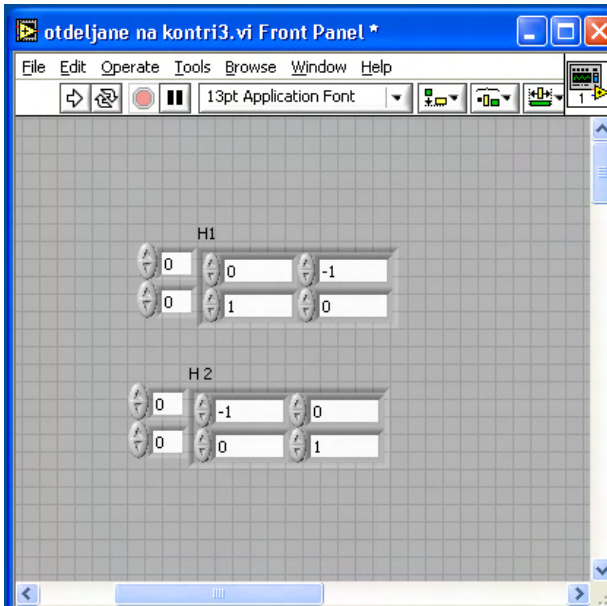


Fig. 2. The Roberts edge operator masks on LabVIEW Front Panel.

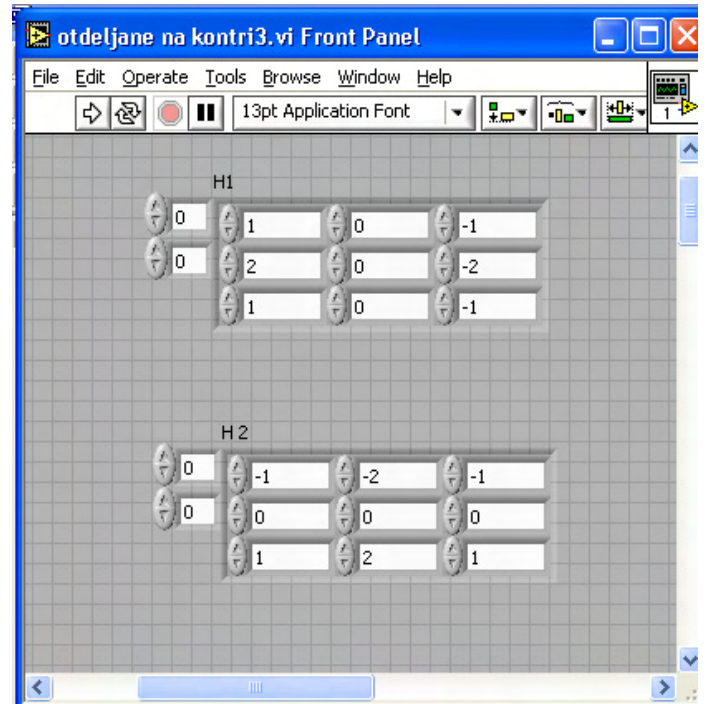


Fig. 4. The Sobel edge operator masks on LabVIEW Front Panel.

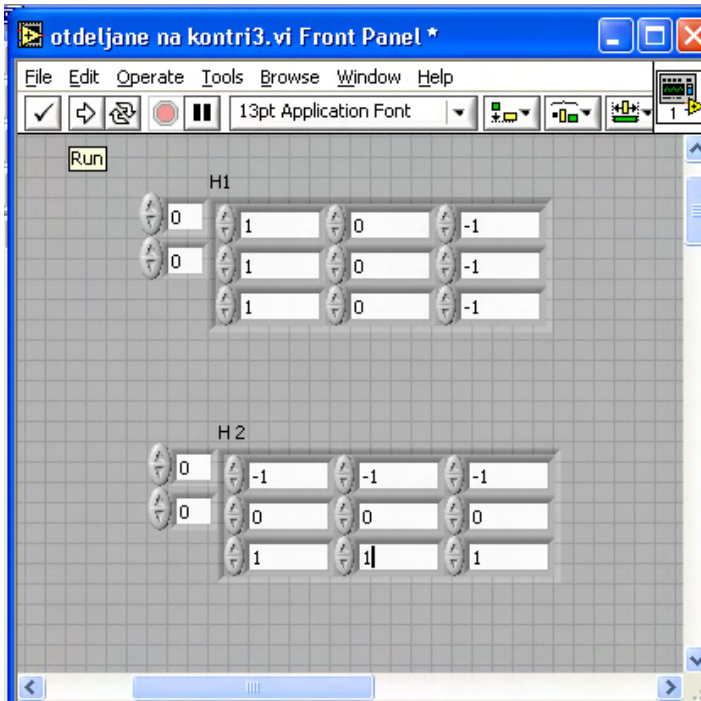


Fig. 3. The Prewitt edge operator masks on LabVIEW Front Panel.

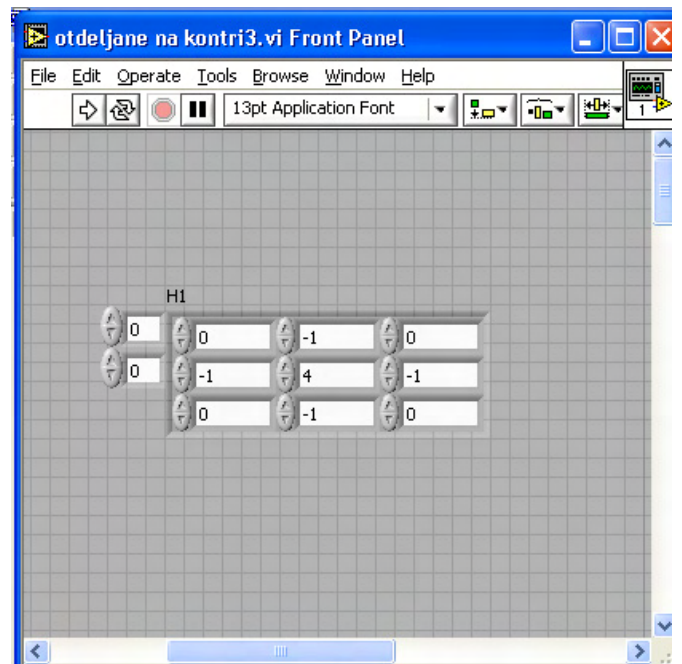


Fig. 5. The Laplasian edge operator masks on LabVIEW Front Panel.

based operator. The Laplasian needs from one mask (Fig. 5) only that simplifies the implementation, but one doesn't provide information about edge direction. This reflects to the LabVIEW block diagram of Laplasian (Fig. 6). It can be seen that one isn't need from computing of magnitude.

It is seen from the LabVIEW block diagram of Laplasian that the decision can be made after applying the mask directly.

IV. CONCLUSION

REFERENCES

In this paper is shown a way to implement four edge detectors using LabVIEW graphical programming environment without any additional tools. The differences in the implementation can be seen. The solution is simple and flexible.

- [1] National Instruments "LabVIEW7 Express. User Manual", National Instruments Corporation 2003.
- [2] E. Nadernejad, "Edge Detection Techniques: Evaluations and Comparisons", Applied Mathematical Sciences, vol.2, pp.1507-1520, no.31, 2008.
- [3] <http://www.cse.uunr.edu/~bebis/CS791E/Notes/EdgeDetection.pdf>

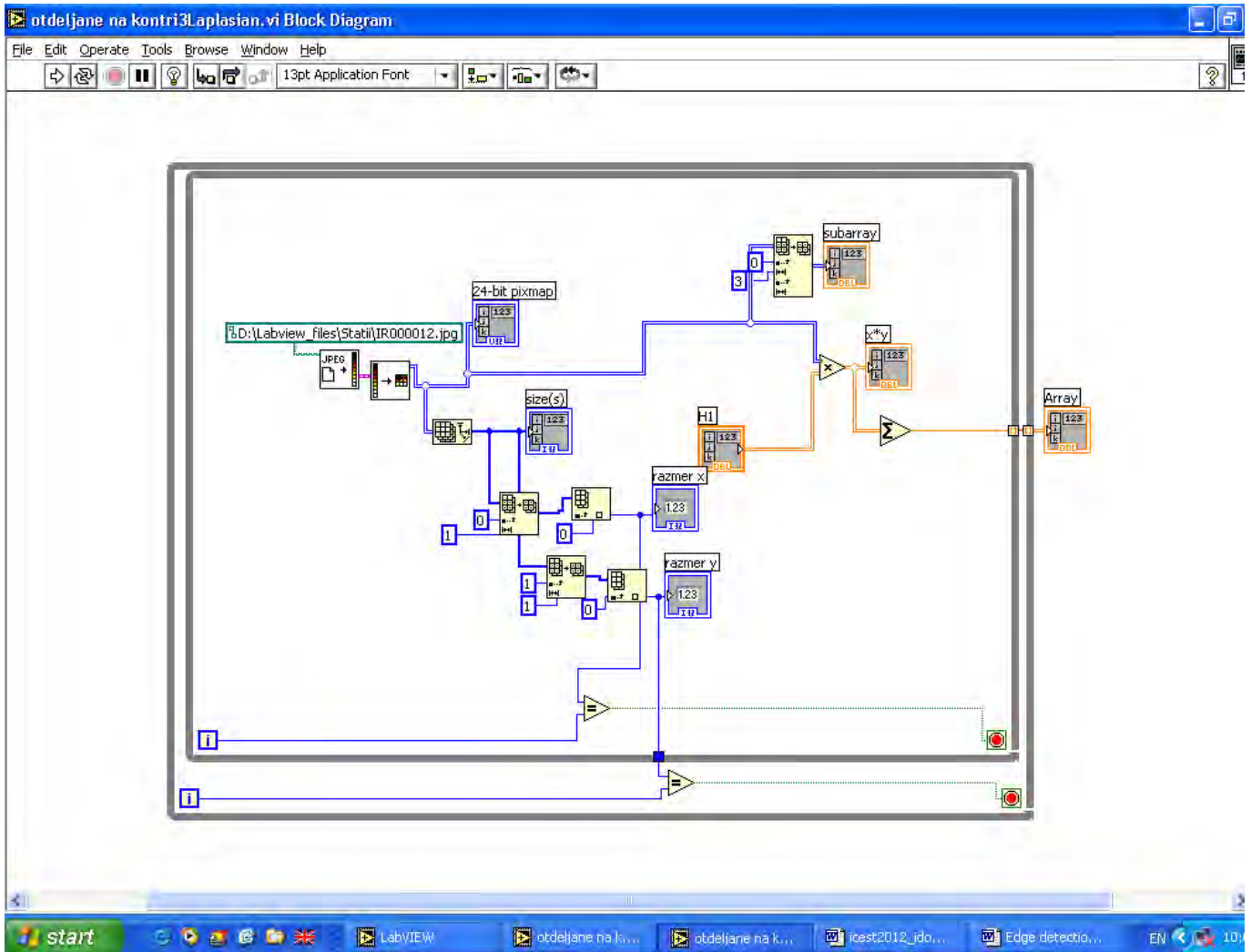


Fig. 6. Implementation of Laplacian based operator with LabVIEW.