Description of image processing algorithms in FPGA for implementation in optoelectronic system in manufacturing

Nikolay Nenov[1]

Abstract – **In the paper the set of image processing algorithms, work in real time, are presented. The proposed algorithms are part ot optoelectronic system for measurement the decentration of optical details and systems during their production. The sequence of processing over the information in the system is described. The features, parameters, used resources in FPGA and way of working of each algorithm are also shown and described.**

Keywords – **Image processing algorithms, CMOS camera, FPGA, Optical measurements.**

I. Introduction

The decentration is displacement of optical axis from the geometrical axis [1]. It is caused from defects in lens surfaces, displacement of the surfaces, or displacement of optical details in optical system. The decentration decrease the image quality sharply and it measurement in optical production is very important. The structure of measurement system is shown on Fig.1

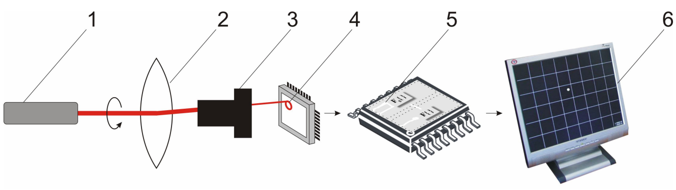

Fig. 1. Structure of the measurement system

The light source 1 emits parallel laser beam which pass though the measured detail 2. The detail is mounted in the precise air bearing and turns around its geometric axis. If the decentration exists, the image of the laser beam in focal surface is not a spot, but circle [2]. The laser image in the focal surface is observed with the lens of the measurement system 3 and CMOS camera 4 [3]. Frames from the CMOS camera are processed in the module for processing of the information 5. Output data are visualized on the monitor 6 or can be sent to the computer for secondary processing and storing.

The frames from the CMOS camera consecutively pass trough the following algorithms:

- Conversion of the BAYER frames from the image sensor into grayscale frames
- Median filtration
- Thresholding
- Calculation of the coordinates of laser spot
- Formation the output data and visualization on the monitor

The common requirements for all algorithms and whole system are: precise calculations; work in real time; possibility for work with signals from CMOS camera; possibility for implementation in FPGA; using of minimum resources.

To meet the requirements, the processing module of optoelectronic system has to have big calculation power and speed. The module for image processing is realized with FPGA Spartan III by XILINX. All algorithms are written in VHDL and simulated with WebPack 6.0 and MathCAD.

II. Description of the algorithms

A. Conversion of the Bayer frames from the image sensor into grayscale frames

First step in processing the frames from color image sensors is color interpolation [4]. In this way the original colors in image are obtained from Bayer matrix. Image processing algorithms work easier with grayscale images. Because of this next step very often is conversion of color images into grayscale images [5]. The proposed algorithm generates grayscale image directly from the Bayer frame obtained from the image sensor.

Green color brings much more information for the human eye. According to this, the green pixels in the Bayer matrix are 50% of all pixels. This feature is used in the algorithm. In output frame, green pixels from Bayer matrix stay unchanged. For Red and Blue pixels the interpolation is used. They are substituted by the mean value of the two neighbor green pixels from the same line in the image - Eq. 1.

$$Gx = \frac{Gn + Gn + 1}{2} \quad (1)$$

where Gn – green pixels from the Bayer matrix;
Gx – Red or Blue pixel

The algorithm is illustrated in Table I.

TABLE I
ILLUSTRATION OF ALGORITHM FOR GREEN INTERPOLATION

X	G0	(G0+G1)/2	G1	(G1+G2)/2	G2
G3	(G3+G4)/2	G4	(G4+G5)/2	G5	X
…	…	…	…	…	…
Gn	(Gn+Gn+1)/2	Gn+1	(Gn+1+Gn+2)/2	Gn+2	X
X	G0	(G0+G1)/2	G1	(G1+G2)/2	G2
G3	(G3+G4)/2	G4	(G4+G5)/2	G5	X

[1]Nikolay Nenov is with the Faculty of Electronic Engineering and Technology at Technical University of Sofia, 8 Kl. Ohridski Blvd, Sofia 1000, Bulgaria, E-mail: nenov@mail.bg.

Red and Blue pixels in first and last column in the frame can not be substituted correctly, because there are no Green pixels in two sides. Because of this, in the output frame these pixels are noted with 'X'.

B. Median filtration

Median filtration in image processing is very important, because it decrease noises in the images and image processing system work better. Common method for median filtration with kernel requires storage of the kernel in memory [6]. The requirements to the system for speed and work in real time make this algorithm inappropriate. The modified algorithm for median filtration was developed. Simulations in MathCAD shown that one-dimensional median filtration can be used with success. The algorithm use data flow from the sensor and equalize three following element in the current line. In this way only 2B memory are used. Used one-dimensional median filtration is described with - Eq. 2.

$$\varepsilon_1(i,j) = med\{\varepsilon(i,j-1), \varepsilon(i,j), \varepsilon(i,j+1)\} \quad (2)$$

where $\varepsilon_1(i,j)$ – pixel in output frame; ε – pixel from the input frame; i,j – coordinates of the current pixel.

Chart diagram of the algorithm for one-dimensional median filtration is shown on Fig. 2.

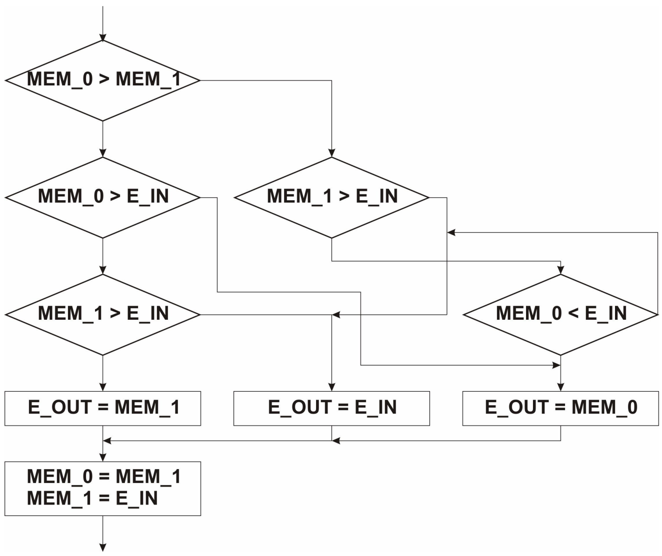

Fig. 2. Chart diagram of the algorithm for one-dimensional median filtration

where E_IN –intensity of the current pixel; E_OUT – intensity of the output pixel; MEM_1 – memorized intensity of the previous pixel; MEM_0 – memorized intensity of the pixel before the previous pixel.

C. Threshold algorithm

Accurate determination of the position of the laser spot requires segmentation of the spot. This is very important for the properly and stable operation of the measurement system. In the machine vision the greylevel of the object differs from the greylevel of the background [6]. Each pixel is classified as either belonging to an object or to the background. The result after threshold algorithm is binary image. In binary image greylevels values greater than threshold have value 1 and belong to the object. Other pixels have value 0 and belong to the background.

If ε is digital image from CMOS camera with M columns, N rows and L greylevels, when the binary image is described with Eq.3. [7]:

$$b(i,j) = \begin{cases} 0, \varepsilon(i,j) < T \\ 1, \varepsilon(i,j) \geq T \end{cases} \quad (3)$$

where b(i,j) is binary image; $\varepsilon(i,j)$ is the intensity of the pixel with coordinates i,j; T, is the threshold level $T \in (0, L-1)$.

The analysis of existing threshold algorithms was accomplished. As an appropriate algorithm, the algorithm of Hamadani was chosen [6]. It is described with Eq. 4, Eq. 5 and Eq. 6

$$T = k_1 m + k_2 \sigma \quad (4)$$

$$m = \frac{1}{M.N} \sum_{i=1}^{M} \sum_{j=1}^{N} \varepsilon(i,j) \quad (5)$$

$$\sigma = \sqrt{\frac{1}{M.N} \sum_{i=1}^{M} \sum_{j=1}^{N} (\varepsilon(i,j) - m)^2} \quad (6)$$

where T – threshold level; m – mean value of the image; σ - standart deviation of the image; k_1, k_2 – constants, depend of the image type.

To be implemented in FPGA, the algorithm had to be modified. In continuous frame generation, the laser spot can move, but in practice during three frames the conditions of the environment stay unchanged. This means that the mean value and standard deviation of the image are the same. For whole calculation of the threshold level three following frames are used. During the first frame the mean value m is calculated. The standard deviation σ is calculated in the second frame using the mean value m from the first frame. Threshold level is calculated in the end of the second frame. Calculated threshold level is used in the third frame. During the second frame, simultaneously with the standard deviation – next mean value is calculated. In this way the algorithm is well suited to the changes in the environment conditions.

The best results was achieved with $k_1 = 1$ and $k_2 = 2$. Chart diagram of the realized threshold algorithm is shown on Fig. 3.

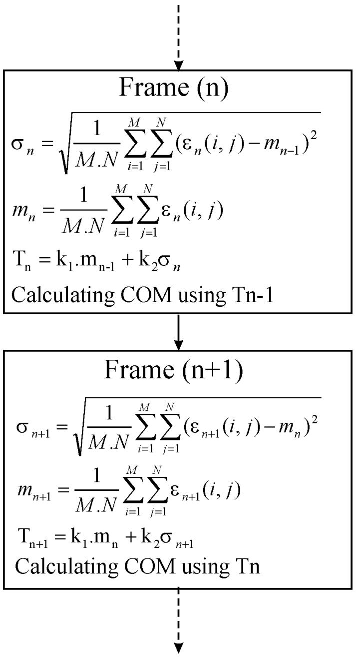

Fig. 3. Chart diagram of the threshold algorithm

D. Calculation of the coordinates of the laser spot

For correct determination of the coordinates of the laser spot, two coordinates X and Y have to be calculated. The best way is the centre of mass to be considered as coordinates of the laser spot – Eq. 7 and Eq. 8.

$$Xc = \frac{\sum_{i=1}^{p} XiEi}{\sum_{i=1}^{p} Ei} \tag{7}$$

$$Yc = \frac{\sum_{i=1}^{p} YiEi}{\sum_{i=1}^{p} Ei} \tag{8}$$

where Xi, Yi – coordinates of the current pixel belong to an object; Xc, Yc – coordinates of the centre of the mass; Ei – the intensity of the current pixel belong to an object; p – number of pixels belong to an object.

The algorithm is modified in following way: Each pixel is checked weather belong to an object (Ei > T). If it belongs, the Eq. 9, Eq. 10 and Eq. 11 are calculated.

$$X1 = \sum_{i=1}^{p} XiEi \tag{9}$$

$$Y1 = \sum_{i=1}^{p} Yi.Ei \tag{10}$$

$$E1 = \sum_{i=1}^{p} Ei \tag{11}$$

In the end of frame, the coordinates of object are calculated according to Eq. 12 and Eq. 13.

$$Xc = \frac{X1}{E1} \tag{12}$$

$$Xc = \frac{X1}{E1} \tag{13}$$

E. Visualization of the frames on screen

For control and set the optoelectronic system and for independent work, the video interface is realized. In this way the images from the CMOS camera can be visualized on a monitor [8].

Consider with the used electronic elements, the achieved parameters of video interface are: grayscale image with SVGA (800x600) resolution and frame rate of 60 Hz. The data is presented with 8 bits, which ensure 256 greylevels. The analog signal for the monitor is obtained by DAC, realized with external resistor matrix with output impedance of 75 Ohm. Used CMOS camera generates 10 fps, but video interface needs 60 fps. Because of the difference between required frame rate, and frame rate generated by the image sensor, additional video memory for synchronization is used. This is the only additional memory whish is used in whole optoelectronic system.

Simultaneously with the image, additional grid scale and coordinates of the laser spot can be shown. This additional information is generated by the “Grid and text generator”, built in the video interface. Chart diagram of the video interface is shown on Fig. 4.

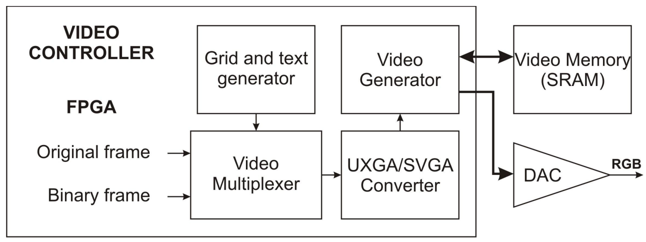

Fig. 4. Chart diagram of the video interface

III. Results

All proposed algorithms have to meet the requirements for speed and have an appropriate bit rate. The bit rate has to be corresponding to the bit rate generated by the image sensor. The CMOS camera generates 10 fps with UXGA resolution (1600x1200). Each pixel is presented with 8 bits. Output frequency is 20MHz (t = 50ns) [3]. The minimum bit rate can be calculated with Eq. 14:

$$C_{min} = \frac{K_b}{T} = \frac{8}{50*10^{-9}} = 152.59 Mb/s \tag{14}$$

where C_{min} – minimum bit rate of the system; T – time for execution of one operation; Kb – length of the binary word.

Used resources in FPGA by the algorithms and achieved bit rates are show in TABLE II. The simulations and results are obtained with WebPack 6.0.

TABLE II
PARAMETERS OF THE PROPOSED ALGORITHMS

Parameter	Algorithms				
	A	B	C	D	E
Slices	32	46	573	513	295
Flip Flops	41	39	429	415	285
LUT	41	76	940	513	513
Max Frequency	194.10 MHz	124,39 MHz	77,90 MHz	68,02 MHz	64,23 MHz
Пропускателна способност	1480,9 Mb/s	948,99 Mb/s	594,35 Mb/s	518,98 Mb/s	490,05 Mb/s

A - Conversion of the BAYER frames from the sensor into grayscale frames
B - Median filtration
C - Thresholding
D - Calculation the coordinates of laser spot
E - Formation the output data and visualization on the monitor

Fig. 5 shows MathCAD simulation of the algorithm for green interpolation.

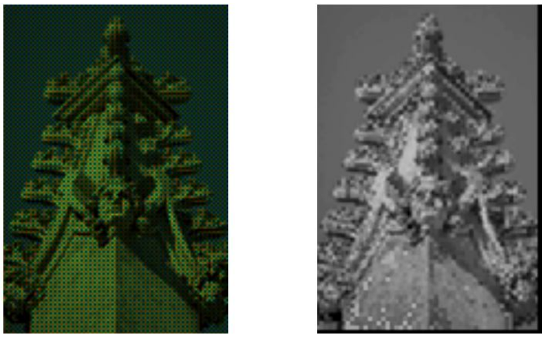

Fig. 5. MathCAD simulation of the algotrithm for green interpolation

In Fig. 6, Fig. 7, Fig.8 and Fig.9 are shown the output frames from each algorithm in the optoelectronic system.

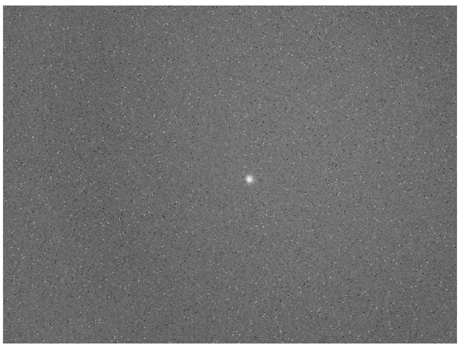

Fig. 6. Frame after green interpolation

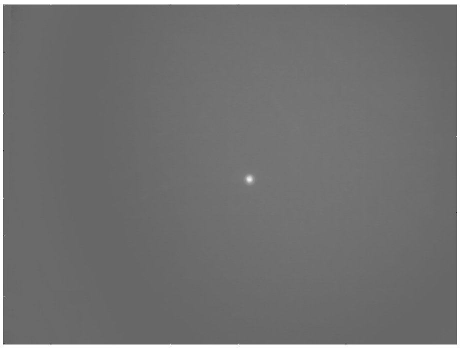

Fig. 7. Frame after median filtration

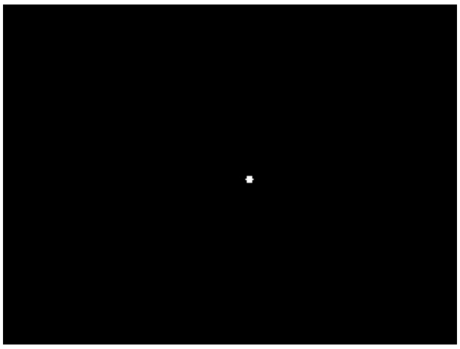

Fig. 8. Frame after thresholding

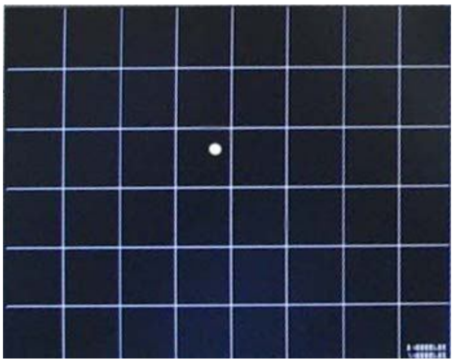

Fig. 9. Frame from the monitor

REFERENCES

[1] Вълева, Е., "Техническа оптика" част 2, Техника, София, 1993, ISBN 954-03-0264-1
[2] Melles Griot - www.mellesgriot.com/pdf/CatalogX/X_20_6-7.pdf
[3] OmniVision Technologies Inc., "OV2610 Color CMOS UXGA (2.0 Mpixel) CameraChip", v1.5 February 1, 2005
[4] Schreiber, W. F., "Fundamentals of Electronic Imaging Systems", Springer- Verlag, Berlin, Germany, 1991, ISBN 0-387-53272-2
[5] H. Myler, "Fundamentals of machine vision", SPIE Publications, 1998, ISBN-13: 978-0819430496
[6] Gerhard X. Ritter, "Computer Vision Algorithms in Image Algebra": New York. CRC Press, 2001.
[7] Myler H., "Fundamentals of Machine Vision", SPIE Optical Engineering Press, Bellingham, Washington, 1999
[8] Hamblen, J., "Rapid Prototyping of digital systems", Kluwer Academic Publishers, Boston / Dordrecht / London, 2001.