FPGA IMPLEMENTATION OF VIDEO PROCESSING-BASED ALGORITHM FOR OBJECT TRACKING

Dan POPESCU[1], Dinu PĂTÂRNICHE[2]

În acest articol se prezintă un circuit FPGA de urmărire în timp real a unui obiect pe baza analizei cromatice. Pentru izolarea obiectului se utilizează distanța Mahalanobis. Procesul de urmărire este divizat în două etape: segmentarea color şi determinarea poziției. Circuitul FPGA primeşte la intrare un semnal PAL/NTSC. Blocurile de bază (Interfața cu camera, Convertorul de culoare, Blocul referințelor, Blocul de segmentare a culorii, Banca de filtre, Blocul de urmărire, Modulul UART şi Ieşirea VGA) au fost proiectate utilizând limbajul VHLD şi au fost implementate pe un dispozitiv FPGA pentru a respecta restricțiile de timp real.

In this paper we propose a chromatic-based real-time single object tracking circuit implemented on a FPGA. For object isolation process the Mahalanobis distance is used. The tracking process is divided in two distinct stages: color segmentation and coordinate determination. FPGA circuit presented in this paper can track a solid color object in a scene when it receives as input a digitized PAL/NTSC stream. The basic blocks (Camera Interface Module, Color Space Converter, Reference Values, Color Segmentation, Filter Bank, Blob Tracker, UART Module, and VGA Output) where designed using VHDL as well as Verilog and implemented on a FPGA-device in order to meet the real-time constraints.

Keywords: object tracking, color decomposition, Mahalanobis distance, FPGA implementation, object recognition, real time image processing

1. Introduction

Tracking of mobile objects, based on image sequences and pattern matching algorithms, was used in the field of robotics, surveillance, shifting estimation, video compression, automatic sport annotation, and human computer interface [1-5]. Successive samples are compared to a reference model in different frames, and so the object position is estimated. The tremendous computational effort to accomplish this task and the high data volume which is required in the process make this approach very difficult to implement it in real-time applications. Thus, we considered the object color like feature in object recognition. To isolate the object from the background, first it is sampled a

[1] Prof., Faculty of Control and Computers, University POLITEHNICA of Bucharest, Romania, dan_popescu_2002@yahoo.com
[2] Stud., Faculty of Control and Computers, University POLITEHNICA of Bucharest, Romania, dinu.patirniche@gmail.com

reference point set from the surface of the object. One drawback in using color cues in machine vision is the color constancy problem that arises due to the uneven illumination of the scene [2]. In the RGB color space the intensity is distributed on all the three components while in the HSV space the illumination information is stored only in the V (value or brightness) component. Hue (H) and saturation (S) are invariant to uneven illumination ([6], [8]) and therefore they can be considered features for object identification process. The magnitudes of the variances of H and S components are of different order, thus we used the Mahalanobis distance to compute a binary mask. The Mahalanobis distance has been widely used in data mining applications, object classifications, as well as in computer vision applications [2]. We proposed developing a specialized FPGA circuit that computes the Mahalanobis distance from a sampled chromatic set in the logHS space.

Before the beginning of the tracking process, the device needs to sample a chromatic set (64 markers) from the surface of the object that will serve as the reference of the entire tracking process (the learning process). The tracking process is divided in two distinct stages, the first being color segmentation. After object recognition, in the second stage the coordinates of the object in each frame are computed. Color Segmentation uses the HSV color space. The variance of a solid color in the logHS space, as resulted from experiments, has not the same statistical distribution on the two axes. In this case, we recommend using the Mahalanobis distance rather than the Euclidean distance in the object isolation process. Fig.1 illustrates the equidistant points using the Mahalanobis distance.

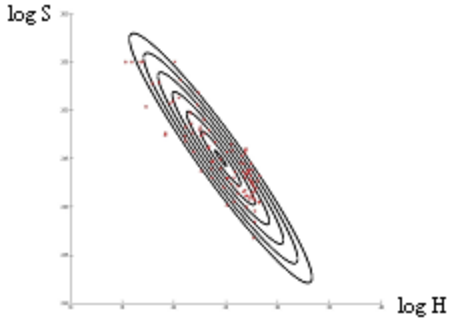

Fig.1. Sampled points from a solid color in the logHS space overlaid with the equidistant lines computed using the Mahalanobis distance

The Mahalanobis distance is defined as:

$$D = \sqrt{(x-\mu)^T \cdot C^{-1} \cdot (x-\mu)} \tag{1}$$

where x is a multidimensional vector $x = (x_1, x_2 \ldots x_N)^T$ which distance is determined, μ is the vector of the means of the reference on all axes and C is the covariance

matrix which resembles the statistical distribution of the reference points in the chosen space. For the bi-dimensional space log*HS*, $(x\text{-}\mu)^T = [H - E(H),\ S - E(S)]$, and C is defined as:

$$C = \begin{pmatrix} \sigma_h^2 & \mathrm{cov}(h,s) \\ \mathrm{cov}(s,h) & \sigma_s^2 \end{pmatrix} \quad where, \tag{2}$$

$$\sigma_h^2 = E[(H - E(H)^2)]\ , \quad \sigma_s^2 = E[(S - E(S))^2, \quad \mathrm{cov}(s,h) = \mathrm{cov}(h,s) = E[(H - E(H))(S - E(S))]$$

E is the expected value for 64 markers.

The position of a pixel in a frame (object coordinates) is determined by its line (x) and column (y). If $V(x,y)$ is the value of the pixel with the coordinates x and y, then the center of mass - coordinates (x_C, y_C) - in a frame is defined as:

$$\begin{cases} x_C = \dfrac{M_{10}}{M_{00}}, \quad y_C = \dfrac{M_{01}}{M_{00}} \end{cases}, \text{ where } \quad M_{ij} = \sum_x \sum_y x^i \cdot y^j \cdot V(x,y), \qquad i,j = 0,1 \tag{3}$$

2. System implementation

The binary mask obtained after the computation of the Mahalanobis distance is affected by noise. To compute the position and the area of the object more accurately, the noise has to be reduced. The filling of the regions which where left out of the binary mask, due to misclassifications, is performed with the 3x3 morphological operators, erosion and dilation. For erosion, if any of the neighbors has the value 0, the pixel in the middle of the mask will be 0. Dilation is the opposite operation. If any of the neighbors is 1, the center pixel will become 1. Fig.2a and Fig.2b illustrate the two operations graphically.

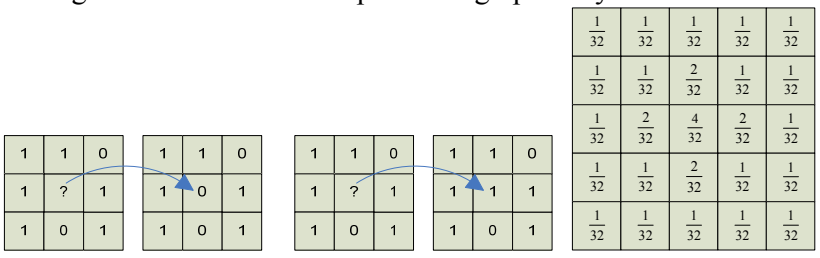

Fig.2. Erosion (a), dilation (b) morphological operators, and smoothing weighted filter (c)

Other similar small regions which result after the segmentation process, and are not a part of the object, are removed with a 5x5 local smoothing weighted filter (Fig.2c). If the sum of all the neighbors is greater than 32, the center will be assigned a 1 value.

The block diagram of the circuit is implemented on the FPGA-device (Fig.3). The basic blocks are as follows: Camera Interface Module, Color Space Converter, Reference Values, Color Segmentation, Filter Bank, Blob Tracker, UART Module, and VGA Output. They where designed using VHDL as well as Verilog and implemented on a FPGA-device in order to meet the real-time constraints.

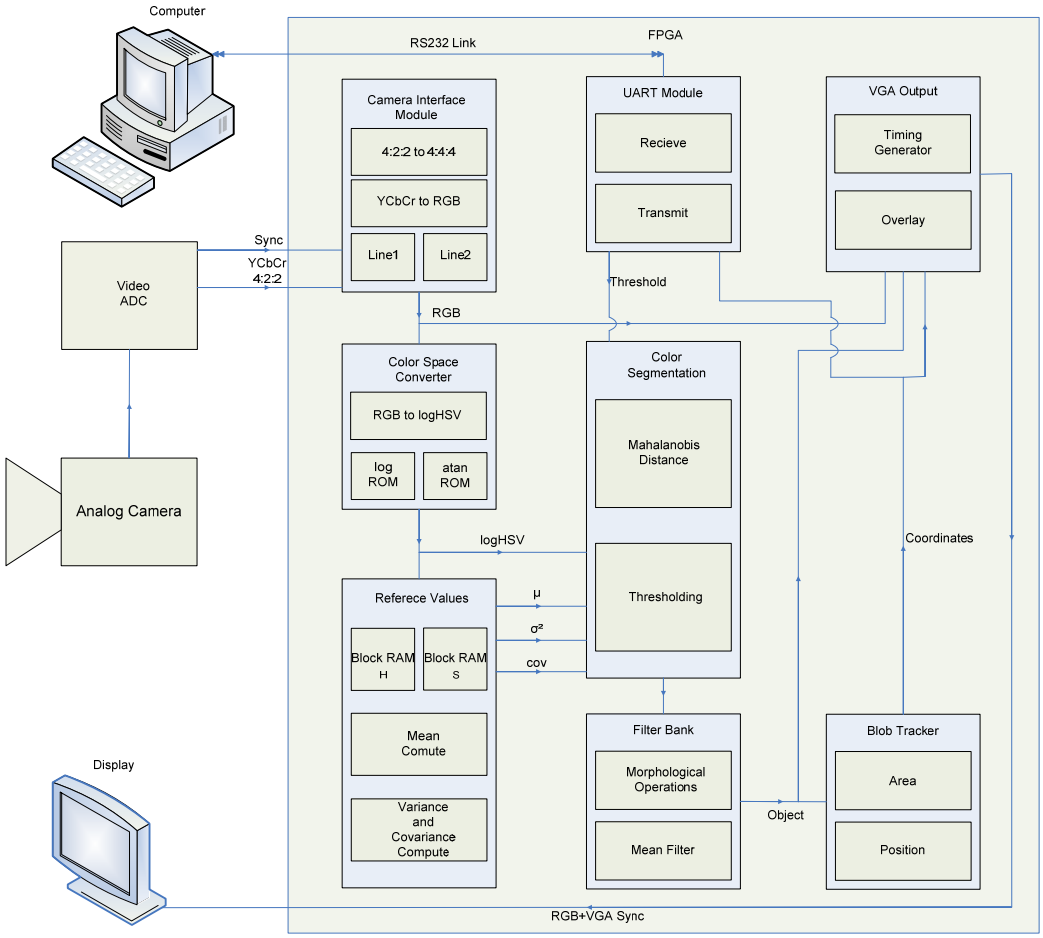

Fig.3. Blocks implemented in the FPGA and the data flow between them

Camera Interface Module

The video stream provided by the color video camera is an analogical signal and before any processing can be done, this signal need to be converted to the digital domain. This task is accomplished by a specialized video ADC that is also capable of decoding the signal. The output of the video processor is a

digitized color information stream (in the YCbCr4:2:2 format) and some synchronization signals (hsync, vsync, pixel clock, field id).

The missing information from the 4:2:2 format is restored using a simple arithmetical mean between two consecutive available chromatic samples, rather than implementing higher order FIR filters. This operation as well as the YCbCr to RGB is performed according to the ITU-R BT.601 standard.

The odd lines of a frame are sent separately from the even lines, the high or low field id signal provides the necessary information to correctly restore the image. Because the frame rate of the display is 60 Hz and the frame rate of the camera is 30Hz de-interlacing can be obtained though line doubling. Two RAM lines are thus present, implemented in the internal block RAM of the FPGA structure. At any moment one of the lines is used to store the incoming data, while the other is used as data output. Inverting is done on the positive transition of the hsync signal.

Color Space Converter

As mentioned earlier, the RGB color space is not well suited for chromatic segmentation of an image because it is very sensitive to illumination changes as well as other factors like rapid movement of the object with respect to the frame rate of the camera. In order to get good object isolation though color segmentation we thus propose the use of the logHSV space instead of the RGB space.

In [8] hue is defined as being the arctangent of a number (4), which means that it can take values between 0º and 360º.

$$H = \tan^{-1} \frac{\log(R) - \log(G)}{\log(R) + \log(G) - 2 \cdot \log(B)} \tag{4}$$

In order to represent H on 8 bits we have chosen to scale the equation (4) with 255/360. We also chose to represent the saturation of a color on 8 bits. Thus, equation (5), which represents the saturation value [10] was scaled with 255.

$$S = \frac{\max(R,G,B) - \min(R,G,B)}{\max(R,G,B)} \tag{5}$$

The computation of the log and arctangent values was not performed on the FPGA; the computational effort would have been too high, so we preferred to implement two ROM-type memories of 256 values each. The values within the ROM tables are scaled as mentioned above, ranging from 0 to 255.

The two divisions where hardware implemented using two fully pipelined dividers. The 8 bit precision combined with the use of the logHS transform was sufficient to accurately isolate the object from its background.

Reference values

Before the tracking of a mobile object can be done, a reference set needs to be provided. The reference set is sampled every time the user sets off a trigger, the values received are stored in the two block RAM memories controlled by this module. Every time a new set is sampled, the five values which describe the set are computed: two means, two variances and one covariance. The means are computed at the first pass though the values. In order to evaluate the variances and the covariance, a second pass is needed.

Because of the parallel structure that is implemented in the FPGA, all three values can be obtained in the same time. Three accumulators are defined with a width of 22 bits. The division is a simple shift to the right with 6 positions, N being 2^6. Multiplications are performed in a single clock cycle using the DSP blocks that are present in the structure.

Color Segmentation

The color segmentation module isolates the object from the background based on the values that characterize the reference set. It relies on computing the Mahalanobis distance from every pixel to the means of the reference set. The distance is than compared to a value that is empirically established, and if it is smaller, in the binary mask a 1 value at the corresponding address is written.

To perform this task in a shorter time, we propose computing the distance in a pipelined manner, rather than describing a data queue and sequentially executing the operations. Squaring equation (1) we obtain:

$$D^2 \cdot \det(C) = (H - \mu_h)^2 \cdot \sigma_s^2 - 2 \cdot \mathrm{cov} \cdot (S - \mu_s) \cdot (H - \mu_h) + (S - \mu_s)^2 \cdot \sigma_h^2 \quad (6)$$

Fig.4 illustrates the pipeline structure which corresponds to equation (6). The pipeline has 5 stages. In order to be passed to the next processing phase the result of a stage is memorized temporarily in a register. DSP specialized hardware blocks where used for every operation, which means they take only one clock cycle to compute the answer, so the entire structure introduces a delay of 5 clock cycles.

Increasing the speed of the processing chain comes at the trade-off of the amount of hardware used (logic blocks). Timing constraints are crucial in real-time applications, and with the increasing capacity of the latest FPGA generations one can implement very large computational pipelines to achieve maximum through output.

Filter Bank

The binary mask obtained after the computation of the Mahalanobis distance is affected by noise. To compute the position and the area of the object more accurately this noise has to be removed. Two types of binary filters are used. One filtering stage was done using morphological operations, erosion and dilation, the other used smoothing weighted mean filters in order to reduce the noise from

the binary image. Morphological operations where performed on a neighborhood of 3x3 pixels [7], while the mean filter had a dimension of 5x5 pixels. All the filtering blocks where connected in cascade and each can be enabled or disabled.

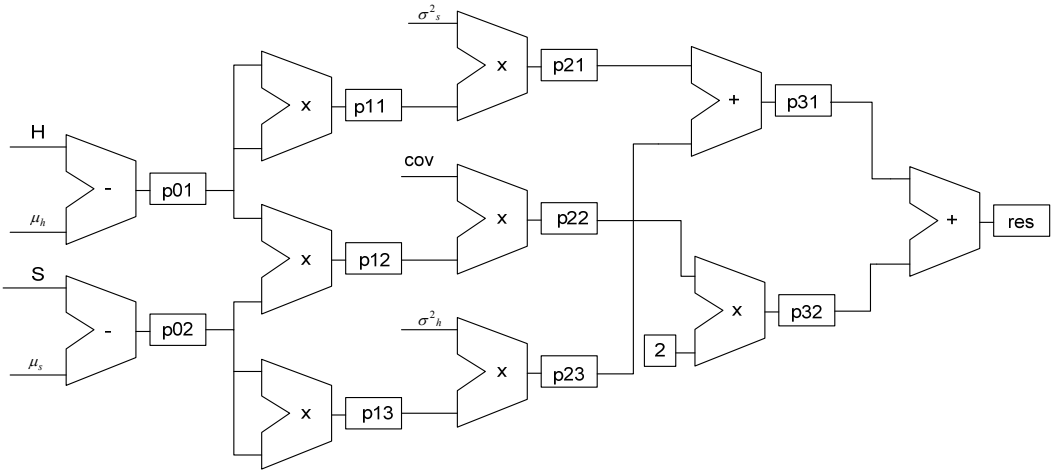

Fig.4. Pipeline structure used to compute the Mahalanobis distance

As mentioned earlier, the result of the color segmentation block is binary, the 3 lines for the morphological operations and the 5 lines for the smoothing mean filter had a dimensions of 720x1 pixels. In both cases one line was used to write the incoming data, the remaining lines provided the values necessary to compute the output of the filter. When the transmission of a new line begins, event marked by the positive transition of the HSYNC signal, the line used to write the incoming data is now used to read values form it and the last line switches from read to write function. The order in which the lines are read is displayed in Fig.5.

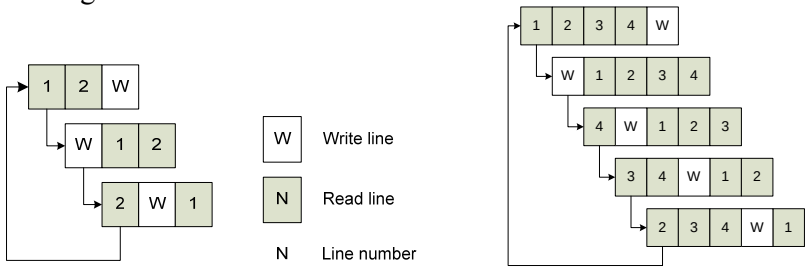

Fig.5. Cycle through the available RAM lines

The morphological operations have a delay of only one clock cycle being computed using continuous assignments though logic blocks (8 input logic AND

for erosion; 8 input logic OR for dilation). The result of the smoothing mean filter is obtained in five clock cycles. The operation is also described in a pipelined manner.

Tracker

Each pixel in the image is identified through three coordinates: line, column, and value. This block computes three parameters of interest: the position of the centroid (two coordinates) and the area that it occupies in the frame. Equations (2) and (3) are used to perform this task. In equation (3), M_{10} is the sum of all line indexes of all pixels having the value 1, while M_{01} is the sum of all column indexes of the pixels with the value 1 [9]. M_{00} is the number of all pixels that correspond to the object in a frame. Three parallel accumulators are used to store the parameters. The division is performed at the end of each frame and implemented using a shared high-radix divider.

UART Module

To have a quantitative measure of the performance of the implemented circuit, the computed values needed to be recorded and analyzed. This operation is performed using a computer, and for simplicity we chose to send the data though the UART protocol at a baud rate of 115200. Also, through the UART module the threshold is sent to the circuit and standard chromatic sets can be loaded into the device.

VGA Output

The display of the video stream and the results of the segmentation and tracking processes are visualized on a VGA compatible monitor. In order to correctly display a video stream, the interface requires only two synchronization signals and the corresponding pixel values. The synchronization signals are generated by the Timing Generator for a resolution of 800x600 pixels. Besides this function, the VGA module overlays a mask in the area in which the object was identified and marks the center of gravity with a cross in order to evaluate the results.

3. Experimental results

The circuit for real-time tracking of mobile object using the logHS space and the Mahalanobis distance was implemented on a Spartan3 FPGA XC3SD1800A-4FGG676C, which has 37440 logic cells, 1800K system gates, 1512 Kb of block RAM, 84 DSP48A DSP units and 519 I/O pins [11]. The FPGA was programmed using Xilinx ISE 10.1 and is capable to acquire a digitized PAL/NTSC video stream, isolate the object, send the extracted coordinates to a computer and display the video stream with a overplayed mask on a VGA monitor. The analog video signal was digitized and decoded using the TVP5147, NTSC/PAL/SECAM 2x10 Digital Video Decoder. Table 1 indicates the summary

of the hardware circuit implemented in the FPGA for the real-time object tracking based on chromatic information.

Table 1

Summary of the FPGA device utilization

Component name	Used	Available	Percent used
Total number of slice registers	3616	33280	10%
Number of occupied slices	2712	16640	16%
Number of 4- input LUT	3625	33280	10%
Number of DSP48A	35	84	41%
Number of RAMB16BWR	36	84	42%

High resolution high frame rate real-time video processing is important in many applications that must provide quick decisions based on events in the scene [12]. The circuit presented in this paper can track a solid color object in a scene when it receives as input a digitized PAL/NTSC stream with a very low input-output latency. The latency is dependent on the frame rate of the video source and the resolution. In our experiment we provided a 720x576@30fps video stream. For this set-up the maximum latency is:

$$\frac{28 lines}{576 \frac{lines}{frame} \cdot 30 \frac{frames}{\sec omd}} \cong 0.0016 \sec onds$$

The tracking object is identified and marked with white (Fig.6). The object unclearness due to the increased speed in the upper right side is diminished by image processing for color segmentation in the bottom right side.

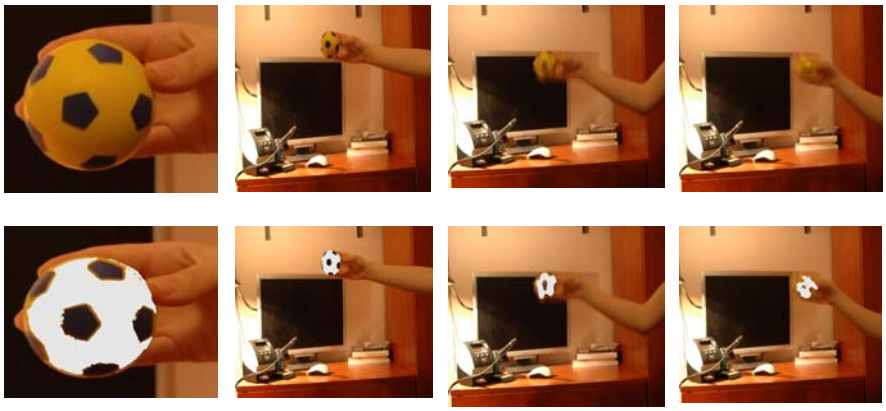

Fig.6. Experimental results of the visual tracking system

4. Conclusions

The real-time single object tracking system based on chromatic information offers a robust, real-time, tracking performance and it is capable of processing digitized NTSC/PAL video stream with a maximum input-output latency of about 0.0016 seconds. Although it provided a video stream with a different clock, it is able to achieve good results. The use of the logHSV space, combined with the computation of the Mahalanobis distance allows the accurate identification of the object in the scene, although it was moved with high speed relatively to the video source frame rate and in an uneven illuminated environment. Because the system was implemented on a low cost FPGA using Verilog and VHDL, it can be used to develop smaller systems with lower cost and high performances. The system was developed in a modular structure, and therefore it is easy to reconfigure and to adapt to specific needs.

R E F E R E N C E S

[1] *P. Guha, A. Mukerjee, K.S. Venkatesh*, "Efficient Occlusion Handling for Multiple Agent Tracking by Reasoning with Surveillance Event Primitive" Proceedings 2nd Joint IEEE International Workshop on VS-PETS, 2005, pp 49-55

[2] *A. Schmidt, A. Kasinski*, "The Performance of Two Deformable Shape Models in the Context of the Face Recognition", Lecture Notes in Computer Science, Computer Vision and Graphics, 2009

[3] *I. Kolonias, J. Kittler, W.J. Christmas, F. Yan*, "Improving the Accuracy of Automatic Tennis Video Annotation by High Level Grammar", 14th International Conference on Image Analysis and Processing Workshops, 2007, pp 154-159

[4] *Griselda Saldana, M. Arias-Estrada*, "Real Time FPGA-based Architecture for Video Applications", IEEE International Conference on Reconfigurable Computing and FPGA's, 2006, pp 1-10

[5] *S. Ahn, W. Kyun Chung*, "Efficient SLAM algorithm with hybrid visual map in an indoor environment", International Conference on Control, Automation and Systems, 2007, pp 663-667

[6] CIE. International Lightning Vocabulary. CIE Publications 17.4, Commision de L'Eclairage, 4th edition, 1989

[7] *D. Popescu, R. Dobrescu, V. Avram, S. Mocanu*, "Dedicated Primary Image Processors for Mobile Robots", WSEAS Trans. on Systems, Issue 8, Vol.5, August 2006, p. 1932-1939, ISSN 1109-2777

[8] *Graham Finlayson, Gerald Schaefer*, "Hue That is Invariant to Brightness and Gamma", Proceedings British Machine Vision Conference, 2001

[9] *D. Popescu*, Vederea artificială în aplicații industriale, Ed. ELECTRA, Bucureşti 2006.

[10] *John Russ*, The Image Processing Handbook, 5th edition, CRC Press, 2007

[11] Xilinx Inc., Spartan-3A DSP FPGA Family: Complete Data Sheet, available from www.xilinx.com, May 2009

[12] *Jung Uk Cho, Seung Hun Jin, Xuan Dai Pham, Dongkyun Kim, Jae Wook Jeon*, "FPGA-Based Rea-lTime Visual Tracking System Using Adaptive Color Histograms", Proceeding of the 2007 IEEE International Conference on Robotics and Biomimetics, 2007, pp172-177.