BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 59, No. 2, 2011
DOI: 10.2478/v10175-011-0017-y

FPGA and ASIC implementation of the algorithm for traffic monitoring in urban areas

S. SZCZEPAŃSKI, M. WÓJCIKOWSKI*, B. PANKIEWICZ, M. KŁOSOWSKI, and R. ŻAGLEWSKI

Faculty of Electronics Telecommunication and Informatics, Gdańsk University of Technology,
11/12 G. Narutowicza St., 80-233 Gdańsk, Poland

Abstract. This paper describes the idea and the implementation of the image detection algorithm, that can be used in integrated sensor networks for environment and traffic monitoring in urban areas. The algorithm is dedicated to the extraction of moving vehicles from real-time camera images for the evaluation of traffic parameters, such as the number of vehicles, their direction of movement and their approximate speed. The authors, apart from the careful selection of particular steps of the algorithm towards hardware implementation, also proposed novel improvements, resulting in increasing the robustness and the efficiency. A single, stationary, monochrome camera is used, simple shadow and highlight elimination is performed. The occlusions are not taken into account, due to placing the camera at a location high above the road. The algorithm is designed and implemented in pipelined hardware, therefore high frame-rate efficiency has been achieved. The algorithm has been implemented and tested in FPGA and ASIC.

Key words: FPGA, ASIC, sensor network.

1. Introduction

Typically, detecting the moving vehicles is done by inductive loops, passive infrared sensors, magnetometers, microphones, radars or microwave sensors. Also high resolution cameras connected to the monitoring centre using high-bandwidth cables or fibre optic links are often used, but collecting data from remote sensors is usually very expensive as installation and usage are concerned. Careful development of the video detection algorithm and proper hardware/software co-design can make the low power devices capable of estimating the traffic flow. Analysis of the video stream locally by each device can significantly reduce the amount of data which needs to be transmitted.

This paper describes an algorithm intended for the use in an intelligent image processing sensor for traffic surveillance system which uses a single, stationary, constant focal length, monochrome camera to detect moving and recently stopped vehicles. The algorithm works with a low resolution camera, since the detected objects (vehicles) are large enough, in relation to the whole image, and their details are not important for this application. The camera is to be mounted at a location high above the road, e.g. on a street-lamp pole, to reduce occlusions of vehicles and provide a large field of view.

2. Image segmentation algorithm

The image segmentation algorithm transforms the camera image into a binary mask containing moving blobs. The algorithm is aimed to be implemented effectively in hardware, performing simple but robust segmentation of traffic objects. The general diagram depicting the idea of the algorithm is presented in Fig. 1.

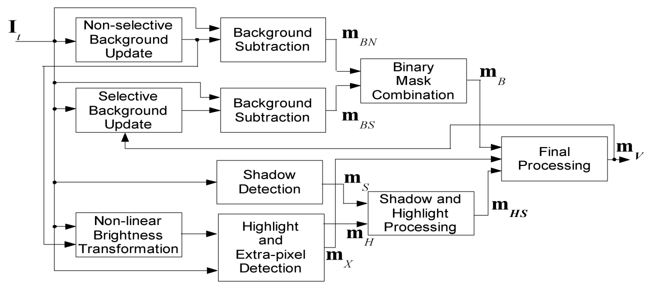

Fig. 1. General diagram depicting the idea of the proposed algorithm

The presented algorithm is based on the background subtraction technique and uses two background models: long-term with non-selective update and short-term with selective background update [1–3]. The models for both selective and non-selective backgrounds are similar; the difference is only in updating the background with data from the current image. For the simplicity of the realisation in hardware, the models assume single Gaussian distribution of pixels' intensities. The pixel of the input image $\mathbf{I}_t$ is classified as foreground using Eq. (1):

$$|\mathbf{I}_t - \boldsymbol{\mu}_\mathbf{t}| > k\boldsymbol{\sigma}_\mathbf{t}, \tag{1}$$

where $\boldsymbol{\mu}_t$ and $\boldsymbol{\sigma}_t$ are mean and standard deviation matrices of Gaussian distribution for image pixel intensities and the constant k typically has a value between 2 and 3. The results are stored as $\mathbf{m}_{BS}$ and $\mathbf{m}_{BN}$ masks for selective and non-selective background, respectively. Detection results from both models have to be combined into a single binary mask $\mathbf{m}_B$. A special combination of *and* and *or* operations is used to improve the detection, similar to the solution described in [3]: when in the proximity of the inspected pixel there is at least one pixel detected by both models, the *or* operation is used, otherwise

*e-mail: wujek@ue.eti.pg.gda.pl

the *and* operation is used. In this way, apart from the current pixel, only the four previously analysed neighbouring pixels are needed. $\mathbf{m}_V$ is the final mask of the blob detection part of the algorithm.

The non-selective background model has longer adaptation times than the selective one, so the recently stopped moving objects are not added to the quickly adapting selective model, because the update is blocked by the mask $\mathbf{m}_V$. After some time, the stopped objects become a part of the non-selective background and then they are quickly included in the selective background, since the mask $\mathbf{m}_V$ stops blocking the update.

The basic detection of shadows in monochrome images can be done simply by comparing the decrease in brightness [4]:

$$m_S(x,y) = \begin{cases} 1 & \text{for } \alpha \le \dfrac{I_t(x,y)}{\mu_t(x,y)} \le \beta, \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

where α and β are constant coefficients: $\alpha = 0.55$, $\beta = 0.95$, evaluated experimentally.

During the night, the illumination of the scene changes drastically. The light reflections from car lights are imposing the detection of many false positive pixels (i.e. the pixels which are falsely detected as the objects). Detection of the highlights, which would work similarly to the solution used in shadow detection, could cause many errors during the day. To solve this problem, the authors propose non-linear brightness transformations, providing different behaviour of the highlight detection block in the day and night. The idea of this method is presented in Fig. 2. The input image $\mathbf{I}_t$ and the background image $\boldsymbol{\mu}_{N,t}$ are first transformed with a non-linear function, which transforms dark pixels into the bright ones and vice versa. For example, a hyperbolic function from Eq. (3) can be used:

$$f(I(x,y)) = \frac{2047}{I(x,y)+1} \tag{3}$$

where $I(x,\ y)$ represents the brightness of the pixel at (x,y), $I(x,y) \in$<0.255>.

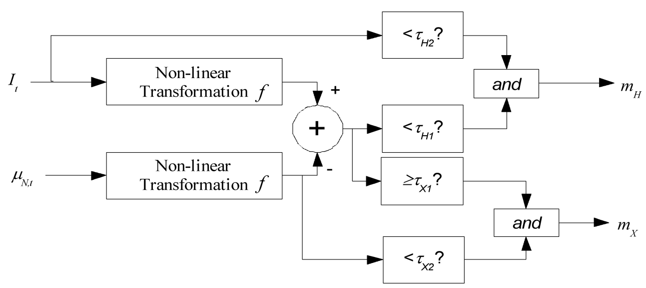

Fig. 2. Flow diagram of highlight detection block

In the night, when the background pixels are mainly dark and are very sensitive to any highlights, after the transformation the difference between the highlight (small value after transformation) and the background (large value after transformation) is large and the highlights can easily be detected and stored as mask $\mathbf{m}_H$. During the day, the difference between the transformed background (large value) and a bright object (also small value after transformation) is smaller than the constant threshold τ_{H1}. An additional threshold τ_{H2} was introduced to exclude very bright pixels from being classified as highlights during the day. Further improvement in the number of true positive pixels can be achieved by detecting very dark pixels on a bright background, also using non-linear transformations. The values of τ_{H1}, τ_{H2}, τ_{X1}, τ_{X2} have to be determined experimentally.

The masks obtained in the previous steps of the algorithm are combined into a single mask $\mathbf{m}_V$ in accordance with Eq. (4):

$$\mathbf{m_V} = (dil(ero(\mathbf{m_H} \vee \mathbf{m_S})) \wedge \mathbf{m_B}) \vee \mathbf{m_X}, \tag{4}$$

where *dil*() and *ero*() denote 2×2 morphological dilation and erosion operation, respectively.

3. Blob analysis and speed estimation

The blobs obtained from the previously described blocks (mask $\mathbf{m}_V$) have to be analysed to detect and to measure the speed of the moving vehicles. Since the camera usually observes the scene at some angle, additional transformations of the image are needed.

For speed estimation, knowledge regarding the relationship between the blobs' dimensions found on the image and the real world coordinates is necessary. Here, the authors assume a model like in [5], where the camera is located above the ground and is pointed towards the road. The ground level is assumed to be planar.

The speed estimation procedure starts with blob image transformation, to obtain linear distances between pixels, proportional to the distances in the reality. Example of such transformation is presented in Fig. 3. Next the detected objects are labelled and the following parameters are estimated for each detected blob: object's boundaries, centre of the object, area in pixels, fill factor (as the percentage of pixels with respect to the bounding rectangle area), etc.

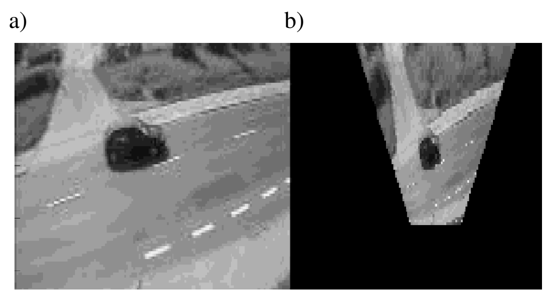

Fig. 3. Example of original image (a) and the result of image transformation (b)

The parameters are calculated using simple operations during pixel by pixel revision of the image. After this stage, the object table with the column number equal to the number of indexed objects is created. Rows describe found parameters. The objects which are too small to represent a moving vehicle are discarded. Such filtering is performed using the object table to remove erroneous blobs. The blobs which are overlapping on two subsequent frames are detected and marked. Such blobs are treated as the same object in movement. Estimation of the speed and direction of the objects is calculated by finding the distance between the centres of the detected objects. For low resolution images with bad quality blobs, speed and direction are calculated using a running average [6].

4. Implementation results

The described algorithm has been implemented in FPGA and ASIC. On-chip implementation included: selective background, non-selective background, highlight, shadow and extra pixel detection, geometrical transformation with blob labelling and blob parameter evaluation. The system has been described in VHDL and implemented using Virtex-4 XC4VLX60 FPGA from Xilinx, the synthesis has been made using Xilinx's XST synthesis software and the implementation has been performed with ISE 9.1 from Xilinx. The ASIC version has been developed using standard cell library FSC0G 130nm from Faraday, the synthesis has been made using Cadence RTL Compiler and the layout has been designed with Cadence Encounter. To decrease the power consumption of the ASIC, the clock gating technique has been used. The ASIC contains DFT versions of the core cells and JTAG I/O chain for testing purposes. The integrated circuit of area of 25 mm^2 was manufactured in UMC 130 nm technology through Europractice. In Fig. 4(b) the picture of the manufactured integrated circuits is shown. Both versions, FPGA and ASIC, are working with 50 MHz clock and process monochrome, low resolution, 128×128 pixels, 32 fps video stream. The measurements show that ASIC realization consumes less power than FPGA counterpart: more than 5 times of reduction for the integrated circuit's core and approx. 3 times reduction for the whole system has been achieved.

To illustrate the operation of the image detection algorithm, the simulation has been made using the artificial test scene of linearly changing background of $\mu_A = 0 \ldots 255$, with rectangular objects casting simulated shadows of intensity $\gamma_k \mu_A$, moving from left to right, as shown in Fig. 5. The intensity of the artificial moving objects are $I_k = \{0, 64, 128, 192, 255\}$, while $\gamma_k = \{0.55, 0.65, 0.75, 0.85, 0.95\}$ for $k = 1 \ldots 5$.

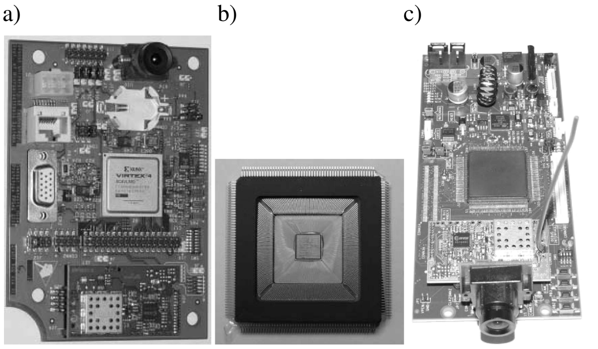

Fig. 4. Photographs of realized circuits: a) FPGA realization of the system running the proposed traffic detection algorithm, b) ASIC with open cover, c) ASIC realization of the system running the proposed traffic detection algorithm

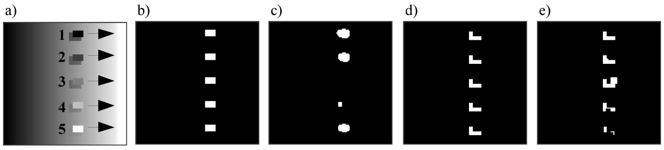

Fig. 5. Simulation results of algorithm operation for artificial scene; a) input image with added objects' indices and arrows indicating direction of movement, b) ground truth for objects, c) object detection results (mask $\mathbf{m}_{\mathrm{V}}$), d) ground truth for shadows, e) shadow detection results (mask $\mathbf{m}_{\mathrm{SH}}$)

The pictures showing several phases of operation of the algorithm are shown in Fig. 6. The prototype modules have been installed on the streets around the Gdańsk University of Technology and tested in the real operation environment. The car detection has been measured for various weather conditions, with a human operator as the reference, counting for 100 vehicles in each test. Depending on the light conditions, the modules were able to correctly detect up to 90% of moving vehicles. During the night the detection quality decreased to 56%, what was mainly caused by the simplified processing of the highlights. The detailed results of the algorithm tests and the FPGA implementation can be found in [7].

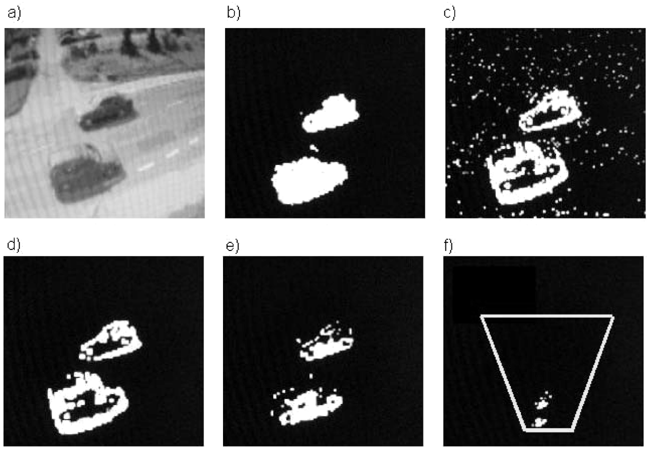

Fig. 6. Selected results of the sensor operation: a) input frame $\mathbf{I}_t$, b) mask $\mathbf{m}_B$, c) mask $\mathbf{m}_S$, d) mask $\mathbf{m}_{HS}$, e) output mask $\mathbf{m}_V$, f) final mask $\mathbf{m}_V$ after the geometrical transformation, the original image boundaries are additionally marked

5. Conclusions

In this paper, the module realizing the algorithm for extracting moving objects from a real-time video stream is presented. The module equipped with camera collects information about vehicles' movement. The steps of the image processing algorithm were carefully selected and adopted to provide simple and straightforward realization in the pipelined hardware. The FPGA and ASIC realizations enabled to practically verify the solution in the real environment.

Acknowledgements. This work was partially supported by the Polish Ministry of Science and Higher Education under R&D grant no. R02 014 01 from the Science Budget 2007–2008.

REFERENCES

[1] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, "Statistic and knowledge-based moving object detection in traffic scenes", *IEEE Proc. Intelligent Transportation Systems* 1, 27–32 (2000).

[2] A. Elgammal, D. Harwood, and L.S. Davis, "Non-parametric model for background subtraction", *Eur. Conf. Computer Vision.* II 751–767 (2000).

[3] D. Duque, H. Santos, and P. Cortez, "Moving object detection unaffected by cast shadows. highlights and ghosts", *Proc. IEEE Int. Conf. Image Processing* 1, 413–416 (2005).

[4] R. Cucchiara, C. Granna. M. Piccardi, A. Prati, and S. Sirotti, "Improving shadow suppression in moving object detection with HSV color information", *Proc. Intelligent. Transportation Syst.* 1, 334–339 (2001).

[5] T.N. Schoepflin and D.J. Dailey, "Dynamic camera calibration of roadside traffic management cameras for vehicle speed estimation", *IEEE Trans. Intell. Transp. Syst.* 4 (2), 90–98 (2003).

[6] D. Koller, J. Weber, and J. Malik, "Robust multiple car tracking with occlusion reasoning", *Proc. Eur. Conf. on Computer Vision* 1, 189–196 (1994).

[7] M. Wójcikowski, R. Żaglewski, and B. Pankiewicz, "FPGA-based real-time implementation of detection algorithm for automatic traffic surveillance sensor network", *J. Signal Processing Systems*, OnlineFirst^{TM}, (2011).