

Fast Prototyping for Video Monitoring Systems with the Use of DSP Module

Agata Chmielewska, Radosław Weychan, Tomasz Marciniak, Adam Dąbrowski, Michał Hartwich,
and Mateusz Owczarczak

Abstract—This paper presents techniques for fast prototyping of real-time hardware / software video processing systems for urban surveillance monitoring equipment. During the experimental research the evaluation module with the TMS320DM6437 signal processor programmed with the use of the Code Composer Studio and Matlab / Simulink environments has been used. Analyzed algorithms can support the work of monitoring video operators. In particular, we analyzed efficiency of implementation of the algorithms using two examples: detection of painting theft and signaling of crossing a pedestrian pass at the red light.

Keywords—Video processing, CCTV system, urban surveillance monitoring, threat detection, Matlab / Simulink, Code Composer Studio, DSP

I. INTRODUCTION

CONTEMPORARY video surveillance systems allow for automatic event detection and analysis of data from video sequences. The main reason for the progress in this area is the use of video processing algorithms and the corresponding human-computer interfaces to facilitate and support implementation of the so-called intelligent monitoring.

The video system capabilities are targeted mainly for remote monitoring of large and overcrowded areas such as: airports, railway stations, industrial plants, or other public areas. For classification of events and behaviors of people in the recorded video sequences the video systems should be assessed with the following issues [1]:

- facilities for acquisition of reliable and representative data without errors of relevant object detection and tracking
- adaptation of algorithms in terms of diversity and complexity of exceptions for correct object recognition
- equipment with adequate models of human behavior without a need for large number of patterns.

Intelligent monitoring systems give a possibility of automatic detection of dangerous situations like: abandoned object detection, pedestrian crossing at the red light, crowd analysis (e.g., at bus/tram stops), tracking of cars driving in the wrong direction on one-way streets, detection of car accidents, fire occurrence recognition, theft detection (e.g., in museums), etc.

This paper presents an approach to fast prototyping of real-time event detection facilities for monitoring systems and provides an overview of simple algorithms that can be used for this purpose in the CCTV (closed circuit television) systems.

Figure 1 introduces the implemented idea of programming the DSP module. It has several advantages. The use of the

All authors are with Division of Signal Processing and Electronic Systems, Poznań University of Technology, Poznań, Poland (e-mails: {agata.chmielewska; radoslaw.weychan; tomasz.marciniak; adam.dabrowski}@put.poznan.pl).

Matlab / Simulink [2] environment enables the simulation verification. The block structure of the project (example presented in Fig. 10) adds clarity and enables the possibility of the rapid algorithm modification. Additionally, the implementation of the algorithm does not require any knowledge of the nuances (often very complex) related to programming of digital signal processor module directly in the Code Composer Studio (CCS) [3] environment with the use of the C language.

For the proper implementation of the algorithms for the TMS320DM6437 EVM (evaluation module) [4], [5] it is necessary to use additional libraries and blocks in the Matlab / Simulink environment that support the connection between the DM6437 and the CCS v. 3.3 software.

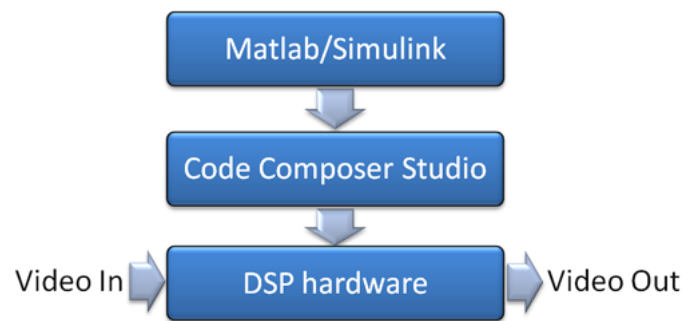


Fig. 1. Schema of fast prototyping with the use of Matlab / Simulink and Code Composer Studio environments and DSP hardware.

The aim of this study was to create systems for detection of dangerous situations for video monitoring. In order to improve the video monitoring operators' concentration, automatic recognition of dangerous situations is necessary. Two algorithms have been presented as illustrative examples. The first one is responsible for detection of the painting theft. The second one is responsible for detection and registration of persons on the pedestrian crossing at the red light [6], [7], [8]. The real-time implementations of both models have been done using the TMS320DM6437 EVM.

II. DSP-BASED VIDEO MONITORING HARDWARE

Figure 2 shows an experimental vision system based on the used Texas Instruments evaluation module. The video signal is transmitted from the camera to the DSP module via composite (or S-video) cable. The processed images can be observed using a standard TV monitor.

TMS320DM6437 EVM is a fixed point unit dedicated to video processing [4]. Analog video data are processed with the use of the TVP5146M2 decoder supporting all typical

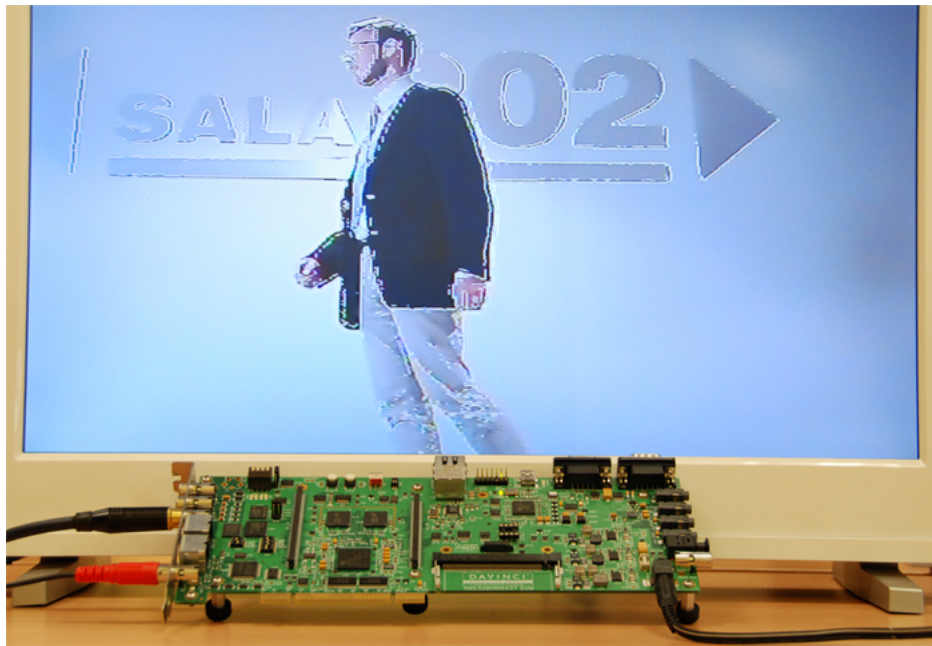


Fig. 2. Example of real-time edge detection algorithm realized with the use of the DM6437 EVM.

video transmission standards within NTSC (national television system committee) and PAL (phase alternating line) standards. The Texas Instruments DM6437 processor is operating up to 600 MHz. It includes eight functional units (two multipliers for 32-bit results and six ALUs (arithmetic logic units), which can produce four 16-bit MACs (multiply-accumulate) instructions per clock cycle, i.e., 2400 MMACs (at 600 MHz). In the case of 8-bit data, it produces 4800 MMACs per second. The EVM is also provided with various kinds of memories: 128 Mbytes of DDR2 DRAM, 16 Mbytes of non-volatile flash memory, 64 Mbytes NAND flash and 2 Mbytes SRAM. Data transfer between EVM module and the PC can be realized with the use of the USB connection or the Ethernet network.

During our tests we have used a typical surveillance camera BOSCH LTC 0455 [9]. The LTC 0455 series cameras are compact rugged, 1/3-inch image format digital color CCD, operating with resolution of 540 TVL. The NightSense mode can be used for expansion of the excellent sensitivity by a factor 3 in the monochrome operation. The video output is provided with the composite video connector.

III. IMPLEMENTATION OF SELECTED ALGORITHM USING MATLAB / SIMULINK

Blocks and functions used to build the real-time software for the monitoring system, which are available in Matlab / Simulink environment thanks the Video and Image Processing library, can be divided into two groups: basic and advanced. Examples of these algorithms are shown in Tab. 1.

A. Motion Detection Model

Motion detection model uses the SAD (sum of absolute differences) function. The acquired sequence is divided into four areas, while the algorithm is used separately for each of them. The detected motion in the appropriate area is highlighted in

TABLE I
 EXAMPLES OF ALGORITHMS AVAILABLE IN THE MATLAB / SIMULINK ENVIRONMENT

Basic	Advanced
Motion detection Edge detection Color segmentation	People tracking Abandoned object detection

red. The SAD value was calculated using formula 1 — it is the absolute value of the difference between the current and the reference image, i.e.

$$SAD(x, y, r, s) = \sum_{j=0}^{15} \sum_{i=0}^{15} |A_{x+i, y+j} - B_{(x+r)+i, (y+s)+j}| \quad (1)$$

where:

$A_{x+i, y+j}$ – pixel values in the active frame
 $B_{(x+r)+i, (y+s)+j}$ – respective pixel values in the reference frame
 $[r, s]'$ – motion vector
 $0 \leq x, y \leq \text{frame size}$.

If SAD values exceed the given threshold, the particular area (the appropriate one from the four mentioned areas) is highlighted in the red color (Fig. 3). The detected movement is visible on the original grayscale image. Each area is assigned to the specified color of the movement intensity chart, which can be seen in Fig. 4. The threshold value is controlled in the Motion threshold block. In the example of Motion detection system the camera is placed in the way that two bottom (from the four) areas (Fig. 3) are positioned on the road. Thus we can easily detect moving cars and their speed. Figure 4 and frames with numbers around 500 show that in the video sequences the

car moved from Area II to Area IV (from the left to the right hand side in relation to the video frame, areas in the image are marked as follows: Area I – left top, Area II – left bottom, Area III – right top, and Area IV – right bottom) — the car movement is visible first in the Area II and latter in the Area IV. The speed of the car was very large (it is visible from the large SAD value). Our threshold is set in such a way that people moving in the top of the video sequence frames are not detected.

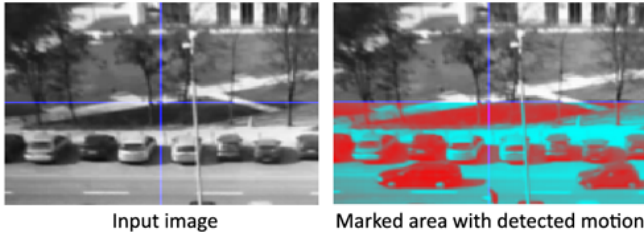


Fig. 3. Results of Motion detection model – the input and output image with the divided detection areas.

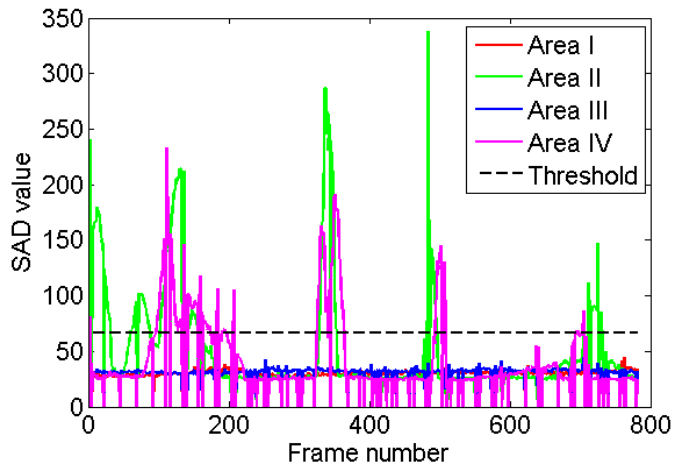


Fig. 4. Results of Motion detection model – the threshold and the SAD chart for the detection areas, where areas in the input image (Fig. 3) are marked as follows: Area I – left top, Area II – left bottom, Area III – right top, and Area IV – right bottom.

B. Edge Detection Model

Edge detection model detects the edges of objects visible in the video sequence. Among algorithms present in the Edge detection block we can select the Prewitt, Sobel, Roberts, or Canny filters. It is possible to change the detection threshold in order to improve the efficiency of the algorithm for particular situations.

During the program operation the image that constitutes the imposition of the input image and the binary image with detected edges are generated. An illustrative result in Fig. 2 shows the algorithm output with the detected edges. In this experiment we used the Prewitt method with 15/256 threshold value and with the edge thinning option enabled.

Research performed on the DSP shows that the algorithm can be evaluated in the real-time for the PAL resolution

(720×576 pixels) with the speed of 4 fps (frames per second), but when we downsampled the original resolution 3 times (up to 240×192 pixels) the video speed could exceed the PAL speed (25 fps) and could be even equal 38 fps.

C. Color Segmentation Model

Color segmentation technique is used to detect and track human faces and hands [10], [11], [12].

The algorithm is based on the image processing, which includes calculation of color to obtain the mean m and covariance C of color components (C_b and C_r).

Color segmentation block calculates the Mahalanobis distance (a distance between two points in a multidimensional space according to equation 2). The result is compared with the threshold value, enabling detection of the skin color.

$$\text{squared distance } (Cb, Cr) = (x - m)' \times \text{inv } C \times (x - m) \quad (2)$$

where:

m – mean

C – covariance matrix

$x = [Cb, Cr]'$.

The result of the algorithm is a binary image, whose pixels with values equal to 1 indicate location of the skin color in the input image. This binary skin map is compared with the original image and the face and hand are detected as marked in the output image.



Fig. 5. Results of real-time face and hand detection performed with DM6473 EVM.

Real-time face and hand detection results performed with the DSP using the Color segmentation model are shown in Fig. 5. This operation (the skin detection) is evaluated with 0.3 fps for the PAL resolution, thus, it is impossible to operate in the real-time for the original resolution. The input image decimation down to (240×192 pixels) gives the result of 3 fps, but the image downsampled to 144×116 pixels allows for fluent image displaying with 8 fps.

D. People Tracking Model

People tracking algorithm is designated to detect and track people in video sequences (Fig. 6) [13]. The object tracking is executed in several stages:

- background estimation with the use of the consecutive 20 frames — Background estimator block
- separation of pixels that represent a moving object and the background — Segmentation block
- pixels connection, which are close to each other (morphological closing operation) and calculation of BLOB (binary large objects) areas (Blob analysis block); adding boundary to the detected people — Tracking block
- estimation of people location based on their location in previous frames — Detection block.

The algorithm is designated to operate on sequences with 160×120 resolution on TMS320DM6437 EVM with 8 fps. In order to operate on the video sequences with higher resolution, we should change the resolution in the Reshape block, which is located in the Background estimator block, but with our DSP, the algorithm with higher input image resolution (240×192) is performed with with 2.5 fps only.

Real-time example results of the model performed with the EVM are shown in Fig. 7. In this example the People tracking algorithm is executed on the grayscale image but the output data are displayed in the YCbCr color space format on the TV screen.

E. Tracking Using Optical Flow Model

Tracking using optical flow algorithm allows to track objects in video sequences using the optical flow technique. This model constructs binary images based on motion vectors using the threshold and the morphological closing operations. Detected areas with movement are marked with rectangles and presented in Fig. 8 [5].

F. Execution Time Analysis

In case of three models described in this section, execution times have been analyzed for four various input video signal resolutions:

- 720×576 pixels – PAL resolution acquired directly from the camera
- 240×192 pixels – image downsampled 3 times
- 144×116 pixels – image downsampled 5 times
- 80×64 pixels – image downsampled 9 times.

Measurements have been done in the Code Composer Studio 3.3 environment with numbers of clock cycles counted for subsequent models. Numbers of cycles have been divided by 594×10^3 , which is the operating speed of the EVM module. The results are presented in Fig. 9. The OY axis is scaled logarithmic, which proves that changing the image resolution influences the execution time exponentially. Figure 9 shows which model can be performed with the DSP with given resolution in the real-time. For example, the Edge detection model could be evaluated with 38 fps for 240×192 pixels resolution, and People detection and tracking algorithm can be evaluated with 8 fps for 144×116 pixels resolution. It should be noted that the above mentioned models have different computational complexities.

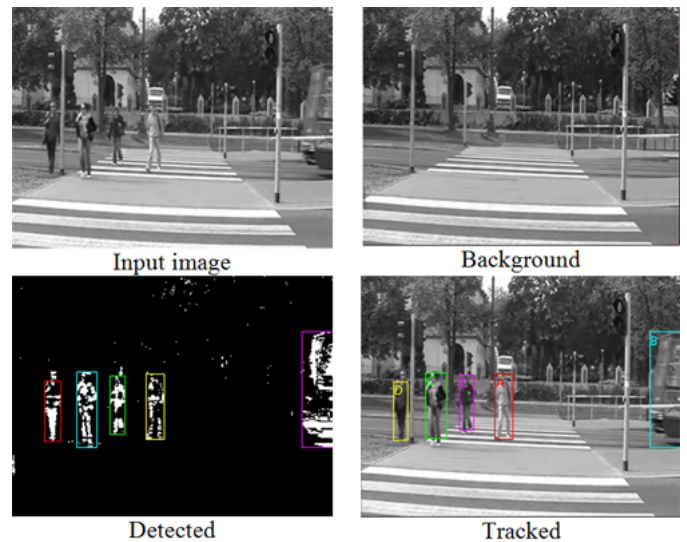


Fig. 6. Results of the People tracking model.

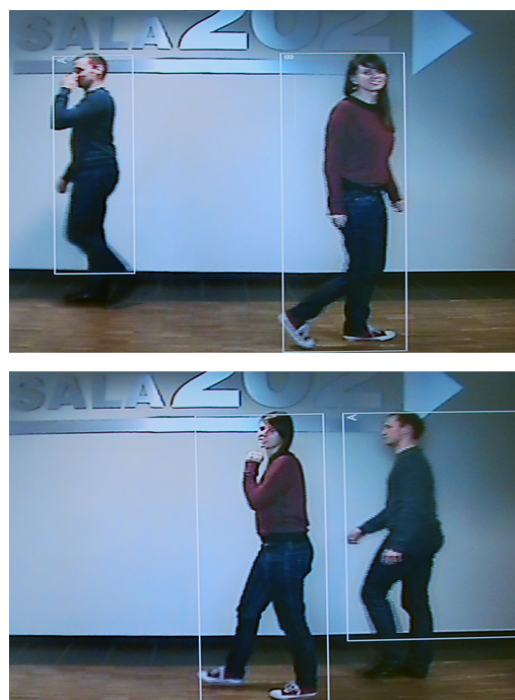


Fig. 7. Real-time results of the People tracking model performed with the DM6437 EVM.

IV. PICTURE THEFT DETECTION ALGORITHM

The research related to the detection of theft is illustrated with the Picture theft detection model. Figure 10 presents this model designed in the Matlab / Simulink environment. The main task of this algorithm is the object disappearance indication from the camera view [14].

The algorithm processes the image in the following steps:

- video capturing from the camera with PAL resolution and YCbCr color space
- the Y color space component from the original video sequence frame is chosen in order to designate the monitored object location — the ROI (region of interest)

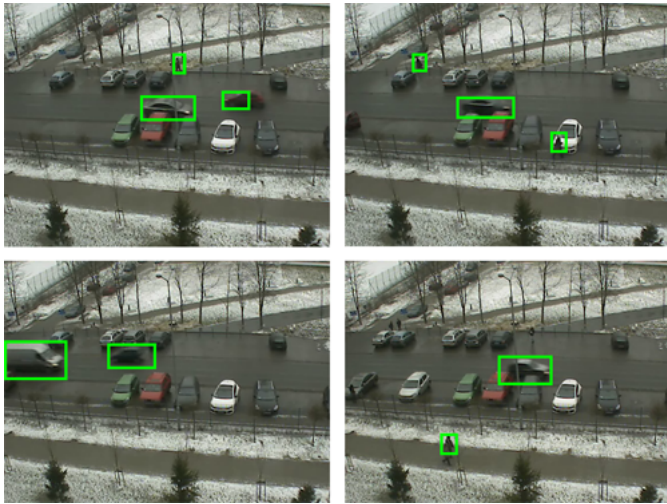


Fig. 8. Results of the tracking using optical flow model.

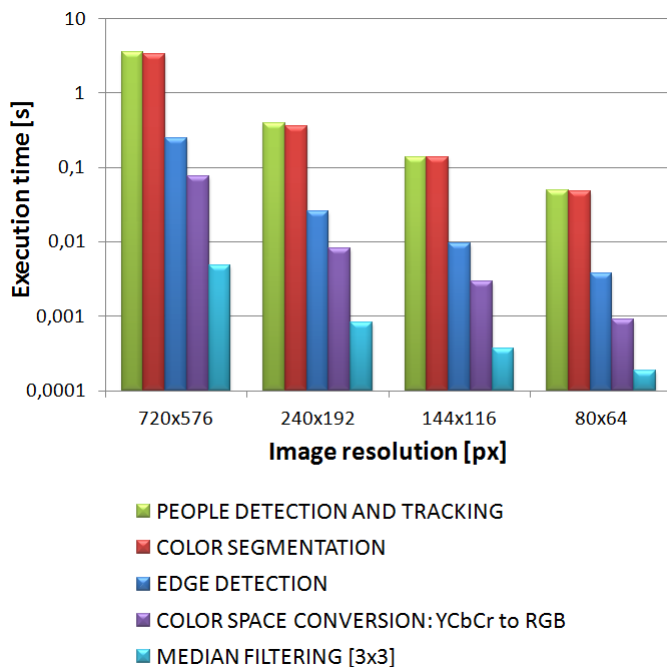


Fig. 9. Execution time analysis for models performed with DM6437 EVM.

- next filtering using the Prewitt mask, which operates for relevant horizontal and vertical edges
- then with the use of the Multiport selector block we choose one from two horizontal picture edges and the morphological closing operation is used in order to improve the edge detection
- at the end stage of the algorithm, the sum of all values is calculated and if this value is smaller than the given threshold, the theft is assumed to be detected.

It should be mentioned that for the correct theft detection, the upper edge of the picture was selected, due to the possibility of covering it by the passing person. Camera should be placed high, near the ceiling; thus people passing in front of the picture do not raise the alarm.

The result of the program is a real-time video with OSD (on screen display) messages located in the top left corner. If the object in the ROI disappears, the algorithm will show the black rectangle in the place the picture should be and in the left top corner of the output image the text: “THEFT – ALARM!” will be inserted. The EVM also enables to send an audio signal; thus there is a possibility of sound alarm emission. In order to verify the presence of the image in the camera view and the proper algorithm operation (when the image is on its place), the text: “PICTURE DETECTED” is inserted.

Figure 12 presents results of the Picture theft detection algorithm. The whole model is presented in Fig. 10.

V. PEOPLE DETECTION AT RED LIGHT ON PEDESTRIAN CROSSING ALGORITHM

Analysis of the detection of moving people through the pedestrian crossing at the red light is based on the model shown in Fig. 11. This model was also developed in the Matlab / Simulink environment. A rule for detecting the red light on the traffic lights is based on the ROI designation and then thresholding and binarization of HSV (hue saturation value) color palette image. In order to provide greater accuracy of moving people detection at the pedestrian crossing, tracking in the appropriate proportions and sizes has been used. For detection of threat, both above mentioned conditions must be fulfilled simultaneously.

A very important issue is the proper camera placement. We decided to place camera 1.5–2.0 m over the street level. An advantage of this decision is that traffic lights hood is not a problem. A drawback is that pedestrians can cover the lights with their bodies and thus it can cause the algorithm failure. However, we found experimentally that this is an extremely seldom situation (it did not happen in our experiments).

The algorithm for better usability and visualization has been additionally equipped with features responsible for informing the system user about changes of the traffic lights. The user is informed about the current color of the light and the number of people counted when standing on the pedestrian crossing while the red light was on. When people pass through the pedestrian crossing in an illegal instant, the algorithm will insert text messages: “RED LIGHT” and “NUMBER OF PEOPLE: x ”. An illustrative screenshot of the user screen is presented in Fig. 13.

VI. SUMMARY

The use of the Matlab / Simulink environment provides a very convenient opportunity for creating, modifying, and testing software based on the C6000 processor family. The presented software and hardware allow the effective implementation of the algorithms such as tracking, classification, and detection of objects in the real-time. This type of programming can be used to realize biometric techniques (e.g., face recognition or iris identification) [15].

Results of multiple tests allow to conclude that the use of the Matlab / Simulink environment and the CCS require the continuous code optimization to implement the system with the highest possible processing rate. Currently, the image

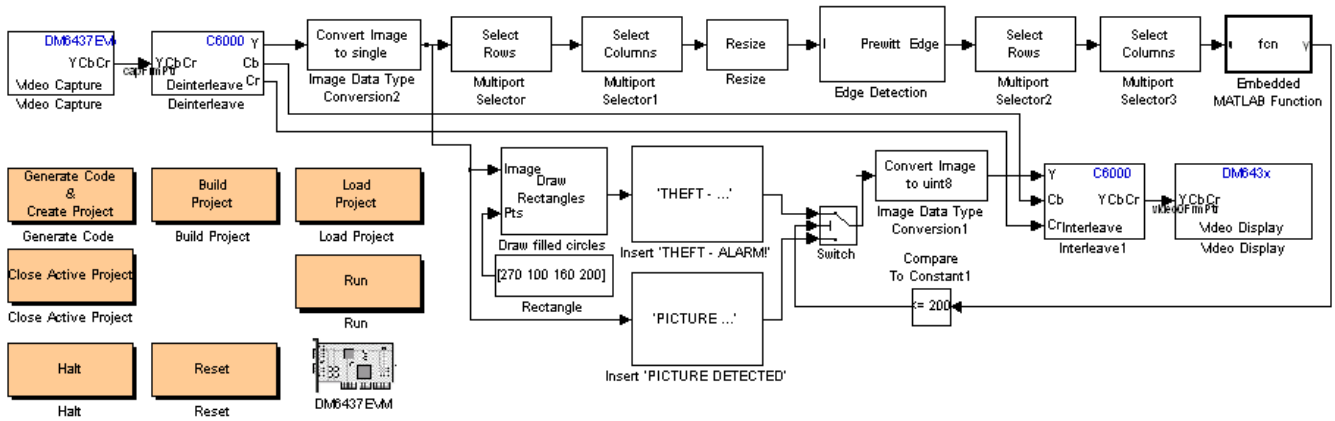


Fig. 10. Picture theft detection model — Matlab / Simulink environment

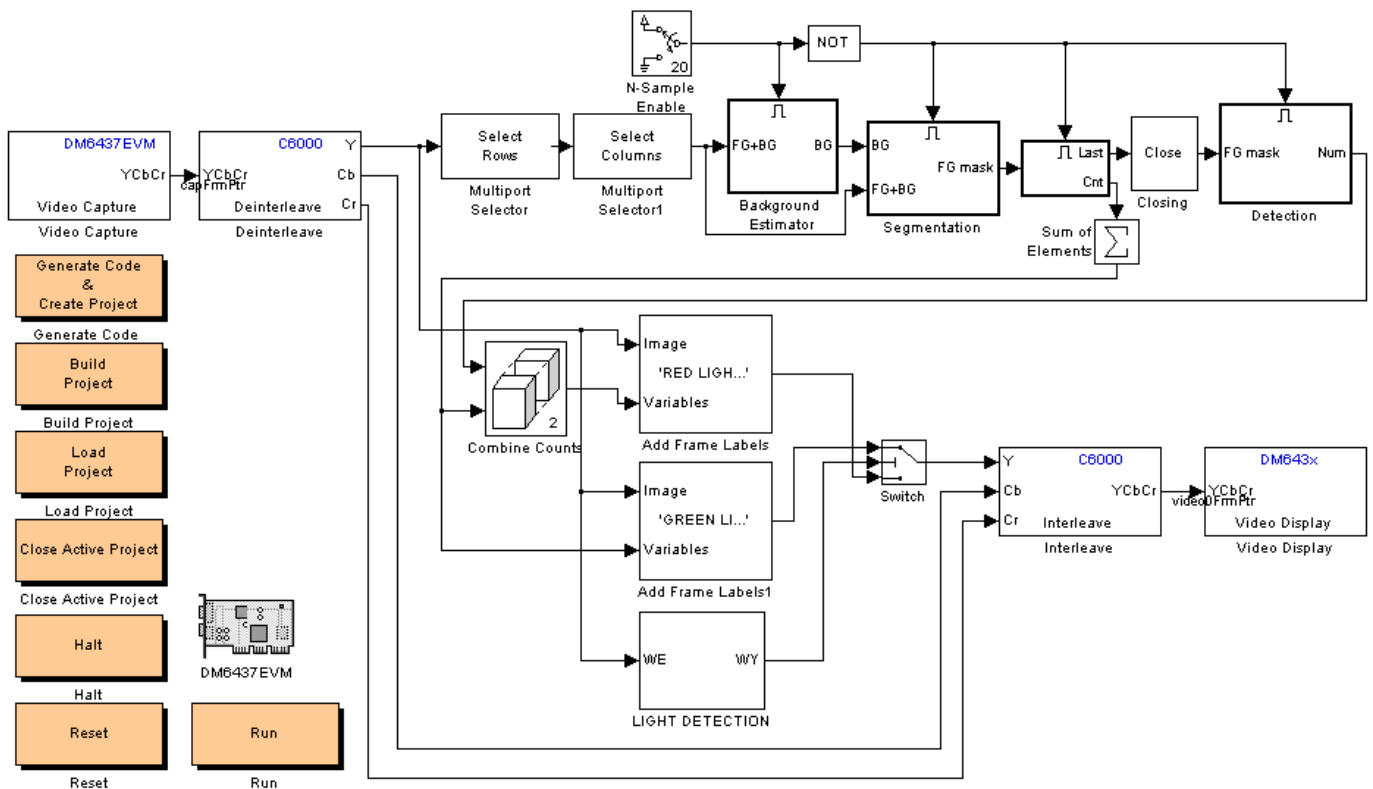


Fig. 11. People detection at red light on pedestrian crossing model — Matlab / Simulink environment

processing within our system runs at 4 fps, which is adequate to the speed of the image registration in contemporary surveillance systems, but it may be too slow in advanced systems working with larger values of fps.

REFERENCES

- [1] D. Kosmopoulos and S.P. Chatzis, "Robust visual behavior recognition," *IEEE Signal Processing Magazine*, vol. 27, no. 5, pp. 34–45, 2010.
- [2] *Matlab/Simulink environment description*, Mathworks Inc., 2013. [Online]. Available: <http://www.mathworks.com>
- [3] *Code Composer Studio IDE Overview*, Texas Instruments, 2013. [Online]. Available: <http://www.ti.com>
- [4] *TMS320DM6437 Evaluation Module with TVP5146M2*, Texas Instruments, 2007. [Online]. Available: <http://www.ti.com/lit/ug/spruev6/spruev6.pdf>
- [5] T. Marciniak, D. Jackowski, P. Pawłowski, and A. Dąbrowski, "Real-time people tracking using DM6437 EVM," in *Proc. IEEE Signal Processing Algorithms, Architectures, Arrangements, and Applications Conference Proceedings, Poland Sect. (SPA'2009)*, Poznań, Poland, Sep. 2009, pp. 116–120.
- [6] J. Balcerek, A. Chmielewska, A. Dąbrowski, D. Jackowski, A. Konieczka, T. Marciniak, and P. Pawłowski, "Recognition of threats in urban areas by means of the analysis of video sequences," in *Proc. IEEE Multimedia Communications, Services and Security (MCSS'2010)*, Kraków, Poland, May 2010, pp. 41–48.
- [7] A. Chmielewska, P. Pawłowski, and A. Dąbrowski, "The use of video sequences processing methods to support the CCTV," *Scientific Papers of Computer Science College in Lodz*, vol. 10, pp. 65–83, Oct. 2010.

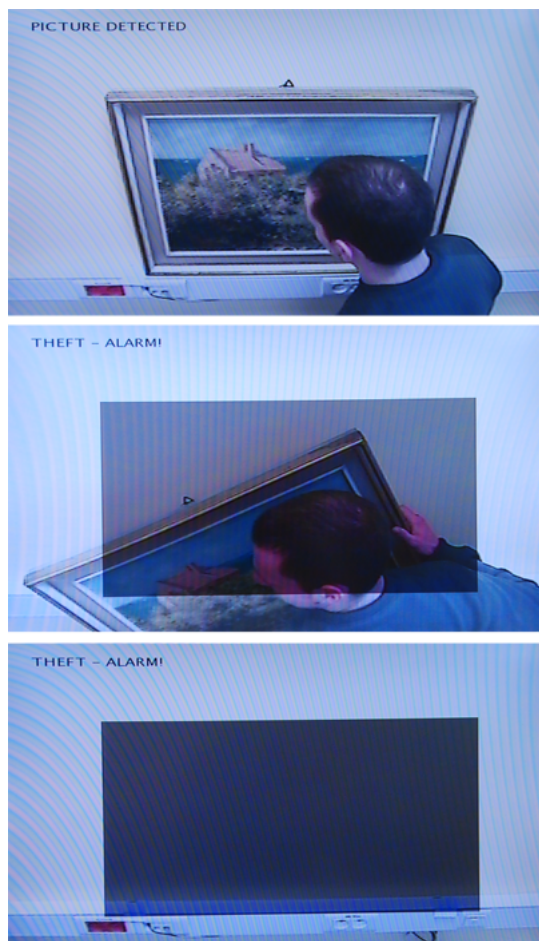


Fig. 12. Results of the picture theft detection model performed with DM6437 EVM.

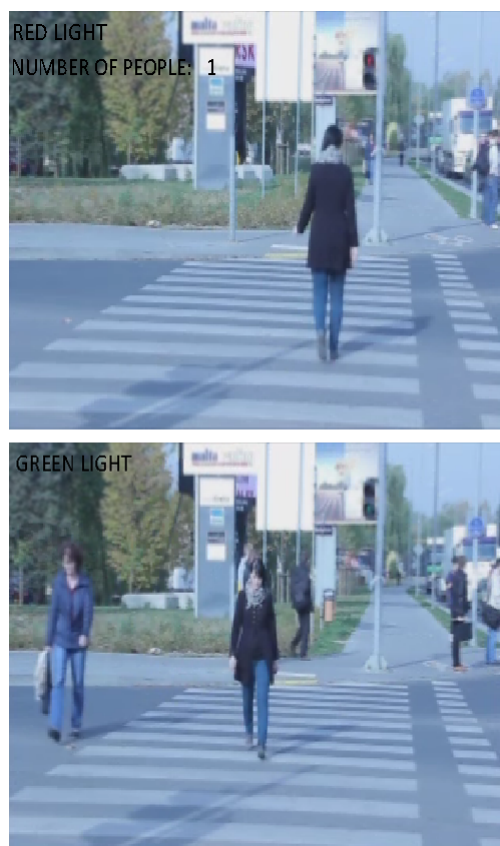


Fig. 13. Results of the people detection at red light on pedestrian crossing model performed with DM6437 EVM.

- [8] T. Marciniak, A. Dąbrowski, A. Chmielewska, and M. Nowakowski, "Real-time bi-directional people counting using video blob analysis," in *Proc. IEEE New Trends in Audio and Video / Signal Processing Algorithms, Architectures, Arrangements and Applications (NTAV/SPA'2012)*, Łódź, Poland, Sep. 2012, pp. 161–166.
- [9] LTC 0455 Series Dinion Color Cameras, Bosch Security Systems, Bosch, 2013. [Online]. Available: http://resource.boschsecurity.com/documents/Data_sheet_enUS_2005022731.pdf
- [10] T. Marciniak, A. Dąbrowski, A. Chmielewska, and R. Weychan, "Face recognition from low resolution images," *Communications in Computer and Information Science*, vol. 287, pp. 220–229, 2012.
- [11] T. Marciniak, R. Weychan, A. Chmielewska, and A. Dąbrowski, "Influence of pose angle on face recognition from very low resolution images," in *Proc. IEEE IEEE New Trends in Audio and Video / Signal Processing Algorithms, Architectures, Arrangements and Applications (NTAV/SPA'2012)*, Łódź, Poland, Sep. 2012, pp. 177–181.
- [12] T. Marciniak, A. Chmielewska, R. Weychan, M. Parzych, and A. Dabrowski, "Influence of low resolution of images on reliability of face detection and recognition," *Multimedia Tools and Applications*, 2013. [Online]. Available: <http://dx.doi.org/10.1007/s11042-013-1568-8>
- [13] P. Pawlowski, K. Borowczyk, T. Marciniak, and A. Dabrowski, "Real-time object tracking using motorized camera," in *Proc. IEEE Signal Processing Algorithms, Architectures, Arrangements, and Applications (SPA'2009)*, Poznań, Poland, Sep. 2009, pp. 110–115.
- [14] A. Pluto-Pradzinski, "Detection of dangerous situations from video monitoring sequences," Master's thesis, Poznań University of Technology, Poznań, 2011.
- [15] T. Marciniak, A. Dabrowski, A. Chmielewska, and A. Krzykowska, "Selection of parameters in iris recognition system," *Multimedia Tools and Applications*, 2012. [Online]. Available: <http://dx.doi.org/10.1007/s11042-012-1035-y>