An Embedded Real-Time Pedestrian Detection System Using an Infrared Camera

Robert Walczyk* **Alistair Armitage**** **T. David Binnie*****

*School of Engineering and Build Environment,
Edinburgh Napier University
email : r.walczyk@napier.ac.uk
***email : td.binnie@napier.ac.uk

** School of Computing,
Edinburgh Napier University
email : a.armitage@napier.ac.uk

***Abstract* — An FPGA-based implementation of a pedestrian detection system using thermal infrared imaging is presented. The main feature of the processing platform is its ability to detect pedestrians at frame-rate from an infrared video stream. It is designed as a stand alone processing unit, supporting security systems in low visibility conditions. The processing unit digitizes a video source from a thermal infrared camera and after pre-processing, provides features of the detected pedestrians such as their position, height and width to a host computer where visualization of tracking and statistical operations can be applied to the received data. The data is transmitted to the host PC via a Bluetooth module interfaced to the development board. The wireless communication allows also for remote control of the processing system's configuration registers. At the present stage of evaluation, the processing unit is limited to detect and transmit the coordinates of one pedestrian in a video frame. The system was implemented using a Xilinx Virtex-II Pro XC2VP30 FPGA chip, no additional memory is required.**

***Keywords* – FPGA, pedestrian detection, infrared.**

I INTRODUCTION

Automated detection and tracking of pedestrians in outdoor environments has many applications in safety, security and people monitoring. Closed-circuit television (CCTV) based systems place heavy demands on computer power and memory, primarily due to the ambient problems encountered in visual pedestrian detection systems. The advent of high resolution cameras and processor intensive classification methods has improved system performance, but placed an even higher requirement on processing power and data storage.

The use of infrared cameras (wavelengths: 8–14µm) for people detection has been common in military systems for some time but camera cost had made the technology largely non-viable for commercial use. Recently low-cost, low-resolution, infrared cameras have become more affordable, allowing the detection and tracking of pedestrians using infrared technology to offer an alternative technical approach to the design and implementation of people monitoring systems. The key difference in the detection of thermal as opposed to visible radiation from a person, is that the thermal radiation is emitted where as the visible radiation is reflected. As the temperature of a human is reasonably constant, this should in principal, make the detection of people less prone to changes in ambient conditions and therefore easier to implement. There are other differences between the representation of a person in a thermal rather than a visible image; not all are advantageous. These significantly affect the technical approach to automated detection design. Thermal images represent heat and have no spectral content. Thermal background is not constant but it is slowly varying compared to changes in ambient lighting which can occur in visible detection. The thermal background can be colder or hotter than the people at a thermal image, termed polarity inversion. Although people stay at a well defined temperature, clothing greatly affects the representation and often leads to disconnected heads and hands. Thermal images are generally of lower resolution and are noisy. Many thermal cameras 'reset' every few minutes, leading to discontinuity in tracking. There can be many other non-human thermal emitters in a scene.

In this paper we report our research into the detection and tracking of pedestrians using low resolution infrared images (LRIRI), and applying FPGA technology for hardware acceleration of our algorithms, with a view to the implementation of a tracking system at frame-rate. The remainder of this paper is organized as follows: Section II gives a short literature review. Section III describes the hardware used in our implementation. Section IV discusses the project's memory requirements. In Section V, the technique used for foreground subtraction is described and evaluated. Section VI presents technique for object detection and tracking; it also explains how the communication between processing platform and stationary PC was established. Section VII shows the experimental results, the last section concludes our investigation.

II BACKGROUND

One of the drivers for early research in automated pedestrian detection using low quality infrared imaging, was the use of a vehicular mounted camera for detecting pedestrians in an urban street at night. Nanda and Davis [1] used a thresholding preprocessor, based on training data, to separate objects of interest from the background. They then applied a probabilistic template consisting of one thousand, 128x48, images which were known to contain people, to identify standing pedestrians in the scene. An experimental dataset showed pedestrian detection rates ranging from 75%-90% with an average of one false alarm per frame and was deemed to be fast enough for practical application. Goubet et al. [2] gives a valuable survey of pedestrian detection technique using IR imagery. The authors use a background subtraction method based on change detection prior to identifying potential pedestrians using morphological filtering and connected component techniques. Importantly they do a qualitative comparison of detection from visible and infrared image sequences using the same scene. Accuracy of foreground (object) detection in the infrared system was highly dependent on threshold level in the background subtraction algorithm. In conclusion they find that classification and tracking is particularly problematic in the infrared due to the variation in thermal radiation from a person due to clothing. More recently Amin et al. [3] compared images from a low cost, low resolution, infrared camera with visible “webcam” images for a people counting application. Using a neural network classifier after thresholding, they found that the results from the infrared image were more accurate for increasing people numbers. Benezeth et al. [4] investigate the application of LRIRI to room occupancy detection. The authors use an adaptive thresholding technique to segment the foreground but rely on the assumption that the foreground will be warmer than the background. They use the intersection of connected components to detect movement but their algorithm cannot cope with occlusion. A statistical model, using 14 features, is used to identify people in the scene. The classifier system does not operate at video frame rate.

A key objective in the design and implementation of a pedestrian detection and tracking system based on image sequence analysis is the achievement of real-time or frame-rate detection. To achieve this, the computational steps involved in the detection and tracking process must be executed in a time less than 40 ms. Hardware acceleration of computational processing stages is a well recognised but technically challenging technique of the speeding up of detection and tracking algorithms. A number or researchers have employed FPGA technology for pedestrian detection in visible image sequences in order to approach real-time detection. Benkhalil et al. [5] report the implementation of segmentation and tracking algorithm on 512x512 visible image sequence using FPGA. The tracking was limited to a single object in the scene. Nair et al. [6] presented an FPGA based people detection system based on the Viola and Jones detection algorithm which operated on compressed visible images sent over an Ethernet which operated at a rate of 2.5 frames per second. Schlessman et al. [7] report a hardware implementation of a software developed tracking system using optical flow algorithm. The dataflow of the system designed as a streaming one with results passed on to the external unit immediately after processing. Kristensen et al. [8] describe an embedded digital video surveillance system. They report implementation of the hardware accelerators for video segmentation, morphological operations and labeling with feature extraction, where tracking was handled in software in an embedded processor core. Their prototype runs at 25 frames per second on 320 x 240 pixel images from Kodak CMOS sensor attached directly to the board.

III SYSTEM OVERVIEW

This section gives an overview of the system designed for pedestrian detection and tracking. A block diagram of the system can be seen in Figure 1. Further subsections provide more details about each component.

a) IR Cameras

The video processing system was tested with two different IR cameras: a ThermoVision Micron Infrared Camera which features a 164 x 128 Indigo VOx uncooled microbolometer sensor array and a FLIR Systems Thermacam PM595 equipped with 320 x 240 uncooled microbolometer focal plane array. The temperature range of the ThermoVision Micron in the standard package is 0^0C to 40^0C, where the scene temperature may be up to 150^0C. This camera delivers performance and features typically found only in larger, more expensive infrared systems. The second camera provided by FLIR Systems has a temperature range of -40^0C up to 500^0C.

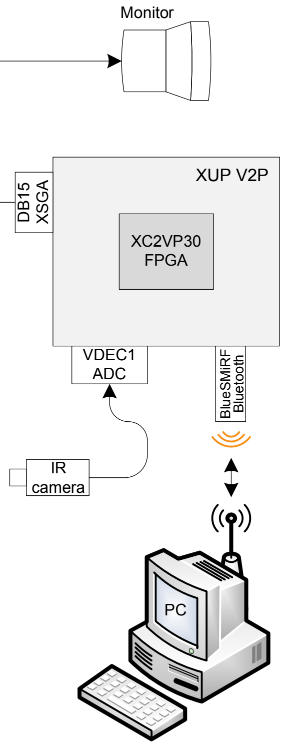

Figure 1: Block diagram of the pedestrian detection system

b) Video Decoder

The VDEC1 Video Decoder Board from Digilent Inc. was used for the analogue to digital video conversion. It employs a ADV7183B Video Decoder chip from Analog Devices. With default settings, it detects and digitizes video source with three 54MHz 10-bit ADCs. The output data is sent to the processing unit in the ITU-R BT.656 format which defines the colour space, the number of samples and sampling format. It is decoded to extract the luminance represented with 8-bit greyscale that is required for further processing.

c) FPGA Development Board

A XUP Virtex-II Pro Development System was used for the processing unit. This is a well equipped FPGA development board with a powerful FPGA chip, widely used for research projects. There is a wide range of peripherals supported by this board making of it an excellent platform for a range of different designs. The Virtex-II Pro (XC2VP30) FPGA chip from Xilinx was used as a base of this board. It features 30,816 Logic Cells, 18 x 18-bit multiplier blocks, two PowerPC processor cores and 2,448 K bits of block RAM which for FPGA chips (comparing to e.g. Spartan-3 family), is a large amount of embedded memory.

d) Peripherals

For output peripherals, both the XSGA output and the BlueSMiRF Gold Bluetooth module from SparkFun were interfaced. Using a standard video monitor display connected to the development board through the DB15 XSGA connector, the system administrator is able to display the processed data as can be seen in Figure 2. This feature was found to be very useful during development work (visual representation of the results for all the processing steps). The video preview of the system results is of key importance during the start-up calibration.

Figure 2: An example output video frame

The communication between processing platform and stationary PC is performed by a BlueSMiRF Gold Bluetooth module attached to the board using one of the available expansion connectors. Essential features of this module are its low power consumption (25 mA avg) and a robust link both in integrity and transmission distance (up to 100 m). For the data transmission a standard UART communication protocol with a speed of 115,200 bps was used.

IV HARDWARE REQUIREMENTS

In order to obtain a real-time pedestrian detection and tracking, the amount of data provided by the video acquisition module had to be considerably reduced. There are only 2,448 Kbits of dual-port Virtex-II Pro Block RAM memory available. In the case of infrared video, both Cb and Cr chrominance components can be omitted; only luma is required in order to represent the greyscale image. Using 864 samples in 625 lines, the digitized video source is decoded giving a representation of 720 x 576 pixels in 8-bit greyscale. To fulfil memory requirements, the video frame is cropped so that only active video area is taken into consideration. An example image of cropped video frame is shown in Figure 3. The final image produced is 640 x 480 pixels. It is not necessary to process at this resolution due to the fact that the array size of both tested IR cameras is much lower. The video source may thus be sub-sampled without the risk of significant loss of quality or information.

The 8-bit greyscale QVGA background model requires 614 Kbits memory. After implementation the image occupies 64 blocks of memory which is 47% of total available integrated FPGA Block RAM. The remainder of the RAM is used for a 4-bit QVGA video buffer as well as for binary results of the foreground segmentation. The final implementation utilises 76% of Block RAM.

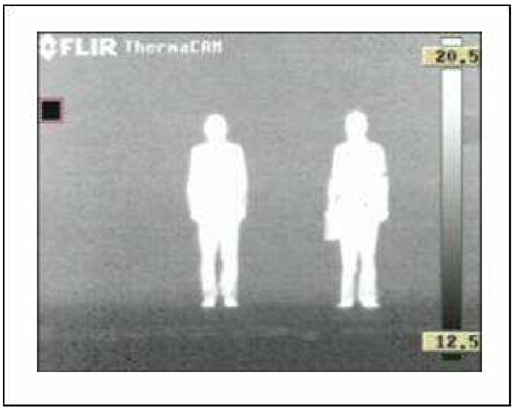

Figure 3: Cropped active video area

V FOREGROUND SEGMENTATION

The first step in image based pedestrian detection systems is a separation of the foreground objects (Regions of Interest) from the background. For this task, a straightforward, memory efficient technique was used. A block diagram can be seen in Figure 4. Background subtraction is performed pixel by pixel simultaneously with data acquisition according to the following equation:

$$\begin{array}{l} \text{if } Source(x, y) > Bckg(x, y) \text{ then} \\ \quad Bckg\_sbtr(x, y) = Source(x, y) - Bckg(x, y) \\ \text{else} \\ \quad Bckg\_sbtr(x, y) = Bckg(x, y) - Source(x, y) \\ \text{end ,} \end{array}$$

where (x, y) are pixel coordinates, *Source* is the present video frame provided by the camera module and *Bckg* is the continuously updated background reference image acquired during the initial calibration. This method supports situations for both pedestrians warmer and colder than the background, *Bckg_sbtr* is the result of the subtraction.

Then an intensity thresholding operation is applied to the image stream resulting from the background subtraction, as follows:

$$\begin{array}{l} \text{if } Bckg\_sbtr\ (x, y) > Th\_level \text{ then} \\ \quad Binary\ (x, y) = 1 \\ \text{else} \\ \quad Binary\ (x, y) = 0 \\ \text{end ,} \end{array}$$

where *Th_level* defines the threshold level which is currently controlled by the system operator using the remote control panel.

As a result of the segmentation step, the data stored in the *Binary* memory array represents pedestrians or other moving objects. As the system detects pedestrians using a thermal infrared camera, any sudden changes in ambient conditions which are problematic in visual pedestrian detection systems, such as light, shadows, rain and clouded conditions, do not add any significant noise to the segmentation unit. An adaptive background subtraction technique was implemented in order to allow the processing system to cope with a constantly changing outdoor, and low light environment. Once the background image is stored onto the FPGA's internal memory during the start-up calibration, the adaptive background subtraction module keeps the background model updated by performing a temporal low-pass filter according to the following equation:

$$\begin{array}{l} Bckg_{i+1}(x, y) = \\ \quad Bckg_i(x, y) + Gain \cdot [Source_i(x, y) - Bckg_i(x, y)] \end{array}$$

The *Gain* factor was found empirically during the experimental work to be preferably equal to 0.125. Additionally, the background should not be updated more often than once a second.

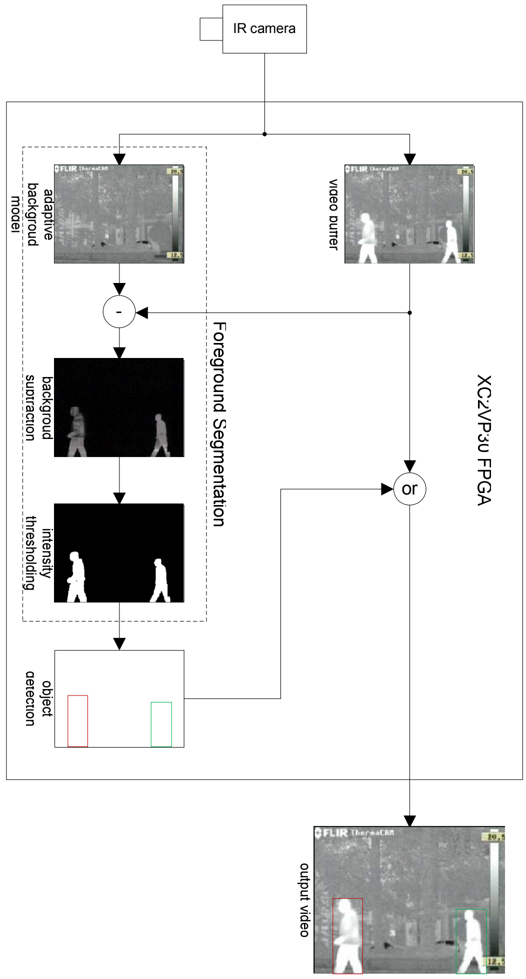

Figure 4: Foreground segmentation data flow

VI OBJECT DETECTION AND TRACKING

The results of the foreground segmentation are stored in the *Binary* memory. Simultaneously with the segmentation step, this memory is scanned in order to find minimum (top left corner - *y_top*, *x_left*) and maximum (bottom right - *y_bottom*, *x_right*) values of coordinates for the region representing a pedestrian. For this task a straightforward scanning technique is used, where the neighbourhood of two consecutive pixels is taken into consideration. At the beginning of each scan, the variables *y_top* with *x_left* and *y_bottom* with *x_right* are assigned with their maximum/minimum values respectively. If two consecutive pixels are found as ones (foreground objects), the following conditions are checked:

- if this is the first line then assign *y_top* with present value of *y;*
- if this is the first appearance in the line then assign *x_left* with present value of *x;*
- similar respective conditions are checked to obtain *y_bottom* and *x_right;*

Simultaneously, other conditions are checked in order to stay with the first object, in case there are other objects detected during foreground segmentation. Although this technique is fast (only one scan is required - no pipelining), at the present stage of the development it gives satisfactory results for only one pedestrian at a frame.

The determined values representing coordinates of the pedestrian are sent in the predefined data frame format to the host PC using BlueSMiRF Gold Bluetooth wireless transmission module. Each package contains six 8-bit vectors as illustrated in Figure 5. As can be seen, the second to fifth bytes represent values of *x_left*, *x_right*, *y_top* and *y_bottom* respectively. The last byte determines the end of the data frame. Due to the fact that there is a need for over 8-bit data representation, the first byte in a package gives an information about MSBs of *x_left* and *x_right*. For testing purposes, measurements were taken and sent to the computer in 100 ms time periods. The limitation for data transmission using the Bluetooth module is about 96 separate targets with measurements taken at a frame-rate (25 fps).

xl[11...8] xr[11...8]	xl [7 .. 0]	xr [7 ... 0]	yt [7 ... 0]	yb[7 ... 0]	control byte

Figure 5: Structure of the data frame

Examples of tracking results are presented in Figure 7 and 7. They can also be shown as 3D plots (Figure 8) where two of the axes stand for horizontal and vertical position of the pedestrian (in pixels), and the third one determines the time in seconds. Figure 6 gives an overview of the control panel. As can be seen, there is a toggle button responsible for establishing a connection. There are three other sections: the first one defines how the video output will be displayed; in the second the tracking time can be set, there is no time limit for tracking, also it is possible to clear the plot. The last one is used during the start-up calibration when the system operator sets the threshold level and stores the background frame in the memory.

VII RESULTS

In this section the key results of the experimental work are reported. The plots shown represent output of the tracking unit. The circle markers show current mid-point of a pedestrian in a single frame, calculated from the coordinates obtained during the detection process. Each line segment connects two consecutive mid-points. Figure 6 shows horizontal pedestrian position against time (seconds). At the 4th second there is some acquisition noise as a person enters the field-of-view (FoV) from the right. This single person is tracked as he/she moves form right to left leaving the FoV at around the 8th second. Between seconds 8 and 14 there are no pedestrians detected. After 14 seconds from the start of the tracking sequence, a pedestrian enters the left side of FoV and moves from left to right.

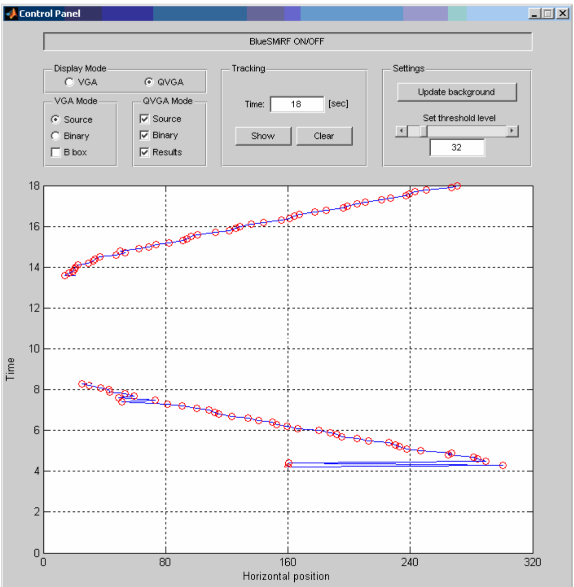

Figure 6: Example pedestrian tracking results

The plot presented in Figure 7 refers to the situation when there are two pedestrians in the FoV both going from right to left. As can be seen, the first pedestrian walks into the scene around the 2nd second and moves out of the FoV around the 5th second, the later part of the plot refers to the second pedestrian who is not acquired until the first pedestrian leaves. Figure 8 illustrates the same information as Figure 7 but shows both vertical and horizontal position in time. We are currently working on tracking multiple objects in real-time.

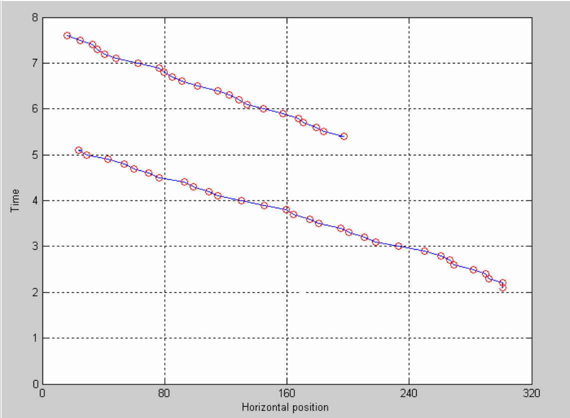

Figure 7: Tracking results: two pedestrians

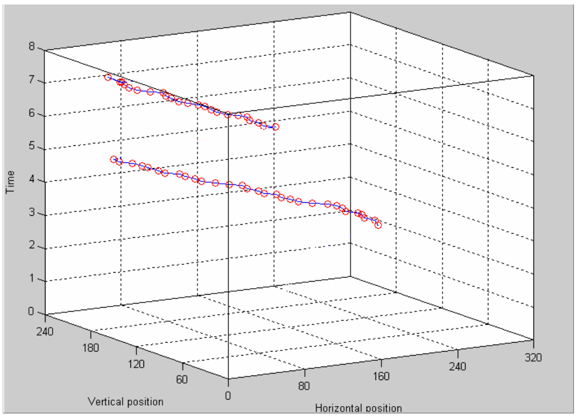

Figure 8: Alternative view for tracking results

VIII CONCLUSIONS AND FUTURE WORK

In this paper we have presented our current work on an embedded real-time pedestrian detection system using infrared imaging. The main feature of the system is that pedestrians can be detected at the full frame rate of the infrared camera (25 fps), a capability difficult to achieve using standard image processing techniques running on even a high-specification PC. Using thermal infrared images confers some natural advantages compared with ordinary visible imaging: pedestrians tend to stand out from the background scene. The use of infrared means that there are some special considerations to the system: firstly, we are operating with the simpler 8-bit intensity of IR images, not full 24-bit colour as might be the case in the visible. Secondly, rather than operating at the typical resolution of a visible camera (for instance VGA) we are operating at the lower resolution of a thermal imager (164 x 128 or 320 x 240 pixels). Finally, we are explicitly making use of the relatively simple nature of the thermal infrared image in which pedestrians appear as relatively bright objects with a significant temperature difference from the background scene.

The use of an FPGA-based system to implement the image processing algorithms means that we can run at the full 25 fps frame rate for signals coming in on a standard RCA or S-Video connector. However, in order to achieve this frame rate, there are some features that differentiate this system from many visible image processing systems. The algorithms employed for background/foreground segmentation are relatively simple: for instance the background model is updated, but only using a low-pass filter. Similarly foreground segmentation uses a straightforward thresholding operation.

The main limitation of the system that we are currently working on is tracking of one pedestrian at a time. In current pedestrian tracking systems working with visible images, sophisticated algorithms are used to identify and label closely spaced pedestrians, and also to resolve occlusion problems. While we do not expect to be able to match the accuracy of the best off-line algorithms, we are working on the labelling to improve the number of pedestrians that can be identified at any one time. In the current system, the location of a pedestrian is shown by a bounding box around the person.

REFERENCES

[1] H. Nanda, and L. Davis, "Probabilistic template based pedestrian detection in infrared videos", Intelligent Vehicle Symposium, 2002. IEEE, vol. 1, Jun 2002, pp. 15-20.

[2] E. Goubet, J. Katz, F. Porikli, "Pedestrian tracking using thermal infrared imaging", Infrared Technology and Applications XXXII. Edited by Andresen, Bjørn F.; Fulop, Gabor F.; Norton, Paul R.. Proceedings of the SPIE, vol. 6206, 2006, pp. 62062C.

[3] I. Amin, A. Taylor, F. Junejoa, A. Al-Habaibeha, R. Parkin, "Automated people-counting by using low-resolution infrared and visual cameras", Measurement, vol. 41, issue 6, Jul 2008, pp. 589-599.

[4] Y. Benezeth, B. Emile, H. Laurent and C. Rosenberger, "A Real Time Human Detection System Based on Far Infrared Vision", Image and Signal Processing, vol. 5099, 2008, pp. 76-84

[5] A. Benkhalil, S. Ipson and W. Booth, "A Real-Time Video Surveillance System Using an FPGA", International Journal of Imaging System & Technology, vol. 11, issue 2, July 2000, pp. 130-137.

[6] V. Nair, P. Laprise, and J. Clark, "An FPGA-based people detection system", EURASIP Journal of Applied Signal Processing, 2005, pp. 1047-1061.

[7] J. Schlessman, C.Y. Chen, B. Ozer, K. Fujino, K. Itoh, W. Wolf, "Hardware/Software Co-Design of an FPGA-Based Embedded Tracking System", CVPR Embedded Computer Vision Workshop 2006.

[8] F. Kristensen, H. Hedberg, H. Jiang, P. Nilsson, V. Öwall, "An Embedded Real-Time Surveillance System: Implementation and Evaluation", Journal of Signal Processing Systems, vol. 52 n. 1, July 2008, pp. 75-94.