A Prototype of Autonomous Intelligent Surveillance Cameras

Wanqing Li  
*University of Wollongong, wanqing@uow.edu.au*

Igor Kharitonenko  
*University of Wollongong, igor@uow.edu.au*

S. Lichman  
*National ICT Australia*

C. Weerasinghe  
*Toshiba, Australia*

Follow this and additional works at: [https://ro.uow.edu.au/infopapers](https://ro.uow.edu.au/infopapers)

Part of the [Physical Sciences and Mathematics Commons](https://ro.uow.edu.au/infopapers)

**Recommended Citation**


Research Online is the open access institutional repository for the University of Wollongong. For further information contact the UOW Library: research-pubs@uow.edu.au
A Prototype of Autonomous Intelligent Surveillance Cameras

Abstract
This paper presents an architecture and an FPGA-based prototype of an autonomous intelligent video surveillance camera. The camera takes the advantage of high resolution of CMOS image sensors and enables instantly automatic pan, tilt and zoom adjustment based upon motion activity. It performs automated scene analysis and provides immediate response to suspicious events by optimizing camera capturing parameters. The video output of the camera can be optimized to any region of interest while the camera continues to monitor the entire scene. Field trials of the prototyped camera have verified the proposed architecture.

Disciplines
Physical Sciences and Mathematics

Publication Details
This article was originally published as: Li, W, Kharitonenko, I, Lichman, S & Weerasinghe, C, A Prototype of Autonomous Intelligent Surveillance Cameras, IEEE International Conference on Video and Signal Based Surveillance (AVSS '06), Sydney, November 2006, 101. Copyright IEEE 2006.
A Prototype of Autonomous Intelligent Surveillance Cameras

Wanqing Li, Igor Kharitonenko
University of Wollongong
{wanqing, igor}@uow.edu.au

Serge Lichman
National ICT Australia
serge.lichman@nicta.com.au

Chaminda Weerasinghe
Toshiba Australia Pty Ltd
cweerasinghe@toshiba.com

Abstract

This paper presents an architecture and an FPGA-based prototype of an autonomous intelligent video surveillance camera. The camera takes the advantage of high resolution of CMOS image sensors and enables instantly automatic pan, tilt and zoom adjustment based upon motion activity. It performs automated scene analysis and provides immediate response to suspicious events by optimizing camera capturing parameters. The video output of the camera can be optimized to any region of interest while the camera continues to monitor the entire scene. Field trials of the prototyped camera have verified the proposed architecture.

1. Introduction

Video surveillance and monitoring systems have become important components in the modern security infrastructure. More and more cameras are installed to provide efficient surveillance, but this also requires employing a sufficient number of skilled personnel for monitoring. According to [1], the most vulnerable part of video surveillance systems is video monitoring personnel. Most of the time, there are no alarming events and the staff may gradually lose concentration on duty. When alarming events are captured by the system, the human response is often delayed and is not optimal.

Another problem that reduces efficiency of video surveillance systems is a contradiction between the required area of view and the sharpness of the captured objects. The wider area captured by a camera, the less resolution it can provide to represent detailed objects. As a result, image quality is usually not sufficient to recognize facial or other important features.

One may conclude that the fundamental problem restricting broader utilization of video surveillance systems is caused by the concept when the system is considered only as an observation and recording device, entirely relied on human attention and decision-making. There is a distinct trend in video surveillance market towards using intelligent systems, which are expected to provide efficient assistance to the operators. The core of these systems is automatic scene analysis, which can be efficiently implemented using distributed cooperative architectures with appropriate level of intelligence at different levels. In this regard, development of technologies and electronic components for intelligent video surveillance cameras that can automatically optimize their parameters and extract critical information for further processing becomes an important practical issue.

This paper presents an architecture and an FPGA-based prototype of autonomous intelligent video cameras that can turn passive surveillance systems into active collaborators to support security operators for immediate and efficient response to suspicious events.

2. Camera Architecture

Figure 2 shows the architecture of proposed intelligent surveillance camera. The camera incorporates an automatic pan, tilt and zoom (PTZ) functionality implemented based upon [2,14] with the parameters decided upon captured activity within a specified region of interest (ROI). This provides more detailed visual information in zoomed modes while monitoring the full view at the same time. The automatic PTZ adjustments can be instantly made based upon motion activity, color and illumination changes of targeted objects.

The camera is built upon a 1.3-megapixel SXGA CMOS image sensor with digital output. Such high image resolution allows displaying Zoom-1 (normal zoom, whole image), Zoom-2 and Zoom-4 without compromising resolution at the video output. This is achieved electronically by cropping a required area of the image sensor just within 40ms. The output video frame from the Color Processing module is 640x256 per PAL field (640x512 interleaved), or digital YUV format for H.263 video codec, depending on the surveillance system architecture. The ROI Module always uses ROI data for monitoring of suspicious events regardless of the currently displayed area.
Moreover, PTZ values are directly used to optimize Color Processing Module parameters in such a way that no specific scaling step is performed before or after the interpolation or color reconstruction of the video frame.

Adaptable image enhancement and noise reduction features are implemented on the interpolated YUV color space within the Image Enhancement module. These functions applied only to the luminance channel (Y), result in significant improvement in image quality, especially while the camera operates in poor lighting conditions.

3. PTZ Based Color Processing

As well known, CMOS image sensors suffer from high level noise and pixel cross-talk [2, 14]. In order to achieve high quality images that are comparable to CCD image sensors, a new color processing chain, as shown in Figure 3, was proposed.

The color correction is performed on the Bayer pattern using a neighborhood-based algorithm. The interpolation algorithm takes into account the correlation among the different color channels and the PTZ values calculated from the currently ROI that is automatically defined by the motion detection and tracking.

The advantage of this color processing chain is that it provides a better quality output with minimum noise escalation using the proposed architecture. However, the splitting of the green channel into separate Gr and Gb channels create non-standard interfaces between the various modules.

3.1 Color Correction

Color correction is performed using an algorithm developed by the authors, which is described in detail in [6]. The Green channel is unaltered. Red and Blue channels are corrected using the following formulae.

\[
R_c = R + 0.5 \times (R - G) \\
B_c = B + 0.5 \times (B - G) \\
G_c = G
\]

The \(R, G, B\) values represent the average channel value computed for each color filter array (CFA) window of size 4×4. The resulting values are clipped to be within the range [0...255].

3.1 Color Interpolation

Color interpolation is performed on the Bayer pattern taking into account the pan tilt and zoom status of the current ROI. This approach can be used both for up and down sampling without performing a specific scaling operation. Many methods are described for down scaling in a CFA pattern by sub-sampling [3]-[5]. However, no methods are described for both up and down sampling of interpolated tri-color data taking into consideration the relative locations of R, G and B components in the captured CFA. This implementation uses a method of weighted bilinear interpolation.
scheme based on relative distance information from the intended interpolated pixel location to the original R, G and B components in the CFA.

The weights are based on the zoom mode and the distance from the intended interpolated pixel location to the original R, G and B components in the CFA. The start addresses for accessing the CFA data are determined by the current pan and tilt values; whereas the address increment rate is determined by the current zoom value. Zoom values 1, 2 and 4 are implemented in the camera [2,14].

The method of address increment together with adaptive weight assignment avoids the usual scale factors associated with conventional scaling. Such scale factors are typically floating point numbers and also involve division operations, which are non-amicable for hardware implementation. The pan (P) and tilt (T) values are also constrained by the current zoom (Z) value as shown in the following equations (1) and (2).

\[
0 \leq P < 1280 \times \left(1 - \frac{1}{Z}\right) \quad (1)
\]

\[
0 \leq T < 1024 \times \left(1 - \frac{1}{Z}\right) \quad (2)
\]

The processing CFA data window is always set to be 4x4 regardless of the zoom value. This is also a hardware friendly feature, since only the most demanding process determines the hardware resources.

In the Zoom-1 mode, CFA frame of size 1280 x 1024 is interpolated to produce a full color frame of size 640 x 512. In the Zoom-2 mode, CFA frame of size 640 x 512 is interpolated to produce a full color frame of size 640 x 512. In the Zoom-4 mode, CFA frame of size 320 x 256 is interpolated to produce a full color frame of size 640 x 512.

4. Image Enhancement

4.1 Sharpness Enhancement

Methods of sharpness enhancement found in literature [7][8] are in the context of compression algorithms (e.g. JPEG), whereas a few deal directly on the luminance (Y) and chrominance (U/V) signals [9].

Under hardware constraints, a simple non-linear and hardware-friendly sharpening filter that proved its efficiency in [2,14] was implemented and employed in YUV color space. The filter has the following properties:

\[ Y' = \left| S_1 - S_{-1} \right| \]

Filter coefficients: If (\( Y' > 20 \))
then { -0.25, -0.25, 2.00, -0.25, -0.25 }
else { 0.00, 0.00, 1.00, 0.00, 0.00 }.

Although the sharpening is performed only on the Y channel, it is important to store the corresponding U and V values to avoid color artifacts. This method performs well for natural images to enhance the appearance of sharpness to the human viewer without significantly increasing the noise level.

4.2 Noise Reduction

The implemented noise suppression algorithm is based on anisotropic diffusion [9,10,11]. Anisotropic diffusion is performed along each video line, for Y channel only. The main reason for this is to minimize the data buffers needed for storing intermediate data between iterations. Only 3 iterations are performed using intermediate storage. Although only Y channel is processed, corresponding U and V values should also be saved, to avoid color shifts at the output image.

A line based anisotropic diffusion algorithm for noise reduction while preserving the edge sharpness was developed and implemented [2,14]. The correction weights for each pixel Y value are based on its immediate horizontal gradient. Consider the following scenario:

\[ Y_{t-2}, Y_{t-1}, Y_t \]

The \( Y_t \) which replaces the \( Y_{t-1} \) is computed as follows:

\[ \nabla_1 = Y_{t-2} - Y_{t-1} \quad (3) \]

\[ \nabla_2 = Y_t - Y_{t-1} \quad (4) \]

\[ Y_t = Y_{t-1} + \frac{\nabla_1}{C_{v1}} + \frac{\nabla_2}{C_{v2}} \quad (5) \]

5. Motion Driven PTZ

Motion analysis and object tracking have been studied for several decades [12,13]. Due to the hardware constrains, a simple and effective algorithm was proposed and implemented for detecting and tracking the motion area to control the PTZ parameters of the camera. Figure 4 shows its flow chart.
5.1 Frame Size Reduction

To reduce the frame captured with 1280 x 1024 pixel resolution, it is subdivided into a number of blocks 64x64 pixels each. Therefore there are 320 (20 x 16) blocks \( B(m,n) \) calculated according to (6).

\[
B(m,n) = \frac{k-1}{k^2} \sum_{x=0}^{k-1} \sum_{y=0}^{k-1} f(kw+x,kn+y)
\]

where \( f(x, y) \) is the initial high-resolution frame acquired from image sensor and \( k \) is the block size, which is equal to 64. In other words, the reduced size frame is obtained by block based averaging of the captured high-resolution frame. Only the reduced size frame (20x16) is used then for motion activity detection.

5.1 Frame Difference Calculation

The frame difference calculation is based on the weighted difference algorithm, which takes into account the block values from several consecutive frames. Such temporal filtering reduces the influence of luminance changes due to the noise.

The difference frame has size 20x16 and consists of the values \( D_t(m,n) \) calculated according to (7).

\[
D_t(m,n) = \frac{3D_{t-1}(m,n) + (B_j(m,n) - B_{j-1}(m,n))}{4}
\]

5.2 Calculation of ROI

If the absolute value of \( D_t(m,n) \) is equal or greater than the threshold \( TH \), then the corresponding 64x64 block is indicated as a motion activity area. Its alarm flag \( A_t(m,n) \) is calculated according to (8).

\[
A_t(m,n) = \begin{cases} \begin{array}{ll} 1 & |D_t(m,n)| \geq TH \\ 0 & \text{otherwise} \end{array} \end{cases}
\]

ROI is defined as the one that includes all blocks, which have non-zero \( A_t(m,n) \) according to (9).

\[
\begin{align*}
m^l_t &= \min(m \mid A_t(m,*) > 0) \\
m^r_t &= \max(m \mid A_t(m,* > 0) \\
n^l_t &= \min(n \mid A_t(*,n) > 0) \\
n^r_t &= \max(n \mid A_t(*,n) > 0)
\end{align*}
\]

where superscripts \( l, r, t, b \) relate to the left, right, top and bottom block boundaries of the ROI respectively.

The zoom factor \( Z \) can be calculated using the following (10).

\[
Z = \begin{cases} \begin{array}{ll} 1, & m^l_t - m^r_t > \frac{W}{2} \text{ OR } n^t_t - n^b_t > \frac{H}{2} \\ 2, & \text{otherwise} \\ 4, & m^l_t - m^r_t < \frac{W}{4} \text{ AND } n^t_t - n^b_t < \frac{H}{4} \end{array} \end{cases}
\]

where \( W \) is the width and \( H \) is the height of blocks within the reduced size frame, \( W=20 \) and \( H=16 \) in our implementation.

The position for panning \( P \) and tilting \( T \) are defined in accordance to (11) and (12).

\[
\begin{align*}
P &= m^c_t - \frac{W}{2 \ast Z} \\
T &= n^c_t - \frac{H}{2 \ast Z}
\end{align*}
\]

where \( m^c_t = (m^l_t + m^r_t) / 2 \) is the central position of the ROI in horizontal direction and \( n^c_t = (n^l_t + n^r_t) / 2 \) is the corresponding center in vertical direction.

Figure 4. A flow chart of the motion detection and object tracking algorithm
5.3 Ballistic Smoothing

Experiments with the intelligent camera [2,14] showed that simple application of the panning and tilting values, P and T, calculated from equations (11) and (12) for repositioning ROI had resulted in a jittered video sequence, especially when several parameters have to be changed at same time. In order to stabilise the sequence, the concept of ballistic smoothing was introduced, which implements a second-order low-pass temporal filter.

The smoothing is applied to the centre position of the detected moving ROI at the original sensor resolution. First, the ROI centre position is converted from block window space to pixel space using equations (13) and (14).

\[ x_c^t = m_x^t * k + (k / 2) \]  
\[ y_c^t = n_y^t * k + (k / 2) \]

Then, the acceleration of the moving ROI (in pixels per frame) is calculated as follows.

\[
\begin{align*}
    a_x^t &= \begin{cases} 
        \frac{y_c^t - y_{c-1}^t}{Dist_{max}} & \text{AND} & \frac{y_c^t - y_{c-1}^t}{Dist_{min}} < s_{t-1} < Sp_{max} \\
        0 & \text{otherwise}
    \end{cases} \\
    a_y^t &= \begin{cases} 
        \frac{x_c^t - x_{c-1}^t}{Dist_{max}} & \text{AND} & \frac{x_c^t - x_{c-1}^t}{Dist_{min}} < s_{t-1} < Sp_{max} \\
        0 & \text{otherwise}
    \end{cases}
\end{align*}
\]

where \( y_{c-1}^t, x_{c-1}^t \) are coordinates of the ROI in the previous frame, \( x_{t-1}^t, y_{t-1}^t \) are the speed of the ROI calculated in the previous frame, \( Sp_{max} \) is the maximum permitted speed of the ROI and \( Dist_{min}, Dist_{max} \) are minimum and maximum allowable travelling distances of the ROI between previous and current frames.

The speed of the ROI (in pixels per frame) is defined as

\[
\begin{align*}
    s_x^t &= s_x^{t-1} + a_x^t \\
    s_y^t &= s_y^{t-1} + a_y^t
\end{align*}
\]

The position of the new ROI is calculated using equations (17) and (18).

\[
\begin{align*}
    x_c^t &= x_{c-1}^t \pm s_x^t \\
    y_c^t &= y_{c-1}^t \pm s_y^t
\end{align*}
\]

\[ y_c^t = y_{c-1}^t \pm s_y^t \] (18)

The coordinates of the new ROI together with the zoom factor are sent to the Colour Processing Module that selects the corresponding area and applies PTZ based color interpolation.

6. Prototype and Results

Figure 5 shows a prototype of the intelligent camera. It was tested in different surveillance scenarios by security professionals. These scenarios include both indoor and outdoor scenes, such as monitoring of the airport air traffic from the distance, monitoring of the vehicles at the highway, surveillance at the exhibition pavilion, hallway and lab monitoring.

Test conditions, test equipment and the obtained results are described in details in [2,14]. The tests showed that camera resolution was up to 800 lines. It had a quick response to the suspicious activity, automatically zooming into the optimal size ROI, which encloses all moving objects and tracking these objects when they move. It should be pointed out that the camera continues monitoring the entire scene while displaying the ROI. If motion activity is detected beyond the current ROI, the camera can automatically adjust the area size. Thus, the objects are captured by the camera with the maximum possible resolution that facilitates further automatic high-level analysis, such as human activity identification [15][16].

Figure 6 shows the hallway tracking sequence when one of the authors walked from the end of hallway towards the camera and then disappeared. The white bars on the right bottom side of the frames show the zoom factor (one bar – for zoom 1, or the whole view field, two bars – zoom 2 and four bars for zoom 4). The blinking red bar in the top right corner of the frames indicates that the motion was detected.

7. Conclusion

The described architecture was implemented on a one million gates FPGA-based platform. Complexity of the FPGA-based implementation indicated...
feasibility of integration of the described solution to any intelligent surveillance systems.

10. References


Figure 6. Hallway motion tracking scenario