An embedded low power high efficient object tracker for surveillance systems

Diaz, Isael; Heijligers, Marc; Kleihors, Richard; Danilin, Alexander

Published in:
First ACM/IEEE International Conference on Distributed Smart Cameras, 2007. ICDSC '07.

DOI:
10.1109/ICDSC.2007.4357546

2007

AN EMBEDDED LOW POWER HIGH EFFICIENT OBJECT TRACKER FOR SURVEILLANCE SYSTEMS

Isael Diaz[†], *Marc Heijligers*[*], *Richard Kleihorst*[*], *Alexander Danilin*[*]

[†]Lund University, Sweden [*]NXP Semiconductors, The Netherlands
[†]sx05id7@student.lth.se, {[*]Marc.Heijligers, [*]Richard.Kleihorst, [*]Alexander.Danilin}@nxp.com

ABSTRACT

This paper introduces a real-time surveillance application for object tracking with capablilities of functioning on 25 f/s mode, implemented in a high efficient, high performance and low power smart camera WiCa (Wireless Camera) developed by NXP Research. The proposed application deals with the problem of constantly changing environments as light change, swaying branches, rain and noise introduced by the camera with techniques of robust background modeling. This article also proposes a method to detect objects merging into the background model named "Forgetting Foreground" Technique and also a new method called "multi-directional image mapping" used for object labeling and segmentation.

Index Terms— Smart Camera, Object Tracking, Embedded, Multidirectional Image mapping, Forgetting Foreground.

1. INTRODUCTION

In Surveillance applications, object tracking plays a very important role when objects or persons are under constant supervision. A complete robust real time tracking system has been a challenging problem for the technological scientific community for the last decade.

Some first attempts to address this problem with a full integrated solution were done around 2000 by Robert Collins [1]. He proposes a complete video and monitoring application, based on a complex distributed network of cameras with a huge and expensive infrastructure. In the same year Comaniciu and Ramesh [2] introduced a not so complete but much simpler solution using mean-shift methods for tracking systems, their solution performs real-time based on features like color or texture. It is partially robust to occlusions, clutter, and small changes in the camera position. It's performance suffers when objects move fast in the scene. In 2004 Khan and Balch [3] introduced a new approach based on eigen-space vectors , this new method presents a good performance but with a very high computational cost. Recently new techniques have been introduced in the same field, Venkatesh Babu and Anamitra Makur offers two completely different solutions. The first uses kernel-computation [4] based on an improved version of the mean-shift tracker, but requires color histograms which are not embedded platform friendly, the second proposed is based on neural-networks [5] requiring a feedback and training not feasible in our platform. In [6] a Markov process is used to model the tracking behavior. In this method candidates of the target are generated in a probabilistic framework color, and motion of the object with motion in the scene. It requires some more resources not available in an embedded platform. IBM Research [7] came into scene with new perspectives on the problem, they have been working in the field with different techniques, they rely on the use of a complex infrastructure where the information is analyzed in computer running specialized software.

We propose a solution integrated into a stand alone camera with embedded video processing capabilities and wireless communication. Our application takes as a initial step a Background Subtraction (BGS) taken from [8] and then a new technique Forgetting Foreground (FFG) is introduced to automatically detect objects standing still in the scene during a certain amount of time, a second novelty is the introduction of Multidirectional Parallel Mapping to extract the images features. This new technique is only achievable thanks to the high parallelism of the platform and the non-linear relation between an image stored in the memory and the SIMD processor, which is the heart of the Smart camera.

The proposed application relies in one of the most modern and high efficient platform today and is able to perform object tracking in real time with no PCs or any extra external processing.

The rest of this paper is organized as follows: In the section 2 we brieftly describe the platform, in the section 3 we describe in detail the methods implemented, the section 4 presents some experimental results and in section 5 we conclude the paper with some discussion for further work.

2. PLATFORM

A smart camera with low power consumption, high processing capability and a tiny size is what the WiCa offers in the smart cameras field, its simple architecture aloud a performance higher than conventional processors. The camera was designed by NXP Research (Formerly Philips Research) [9]

1-4244-1354-0/07/$25.00

for High Demanding computation video processing. The platforms processing capabilities have been proved in [10], [11] Simmons and Ljung built a WiCa network and successfully communicated several cameras by wireless zig-bee connection. In [12] Jeanne and Jegaden implemented a real time face detection using properties of light being projected in the peoples face.

Fig. 1. WiCa (Wireless Camera)

The WiCa achieves a maximum performance when parallel operations are required. Its core is a Single Instruction Multiple Data (SIMD) processor with 320 pixel processors, this processor called XeTal [13] operating at parallel with a embedded RAM memory and I/O interfaces optimized for real time video processing.

The camera is equipped with two CMOS color sensors supporting up to VGA resolution formats; the sensors are separated to each other for some centimeters and this characteristic gives the camera the ability to measure depth with some computational cost.

The WiCa architecture is composed by an 8051 as a master processor, holding a small operating system handling the communication and global operations of the camera, the XeTal as a slave processor for high computation demanding operations, both processors are sharing a Dual Port RAM, which aloud to both processors to operate on the same data.

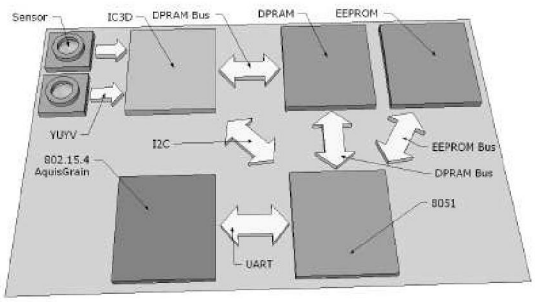

Fig. 2. WiCa Block Diagram

2.1. Sensors

The two image sensors in the WiCa are originally designed for small devices like mobile phone or PDAs. Every sensor is a compact CMOS color camera module with embedded Camera Signal Processor that support up to VGA resolution formats in a small package including a focused optical system.. The device is programmable via an I^2C serial interface. The sensors contain features like Auto White Balance, Auto exposure, Lens shading corrections, 8 bit parallel YUV, or RGB.

2.2. XeTal

The XeTal is the core of the WiCa platform, with a processing power of 50GOPS running at a frequency of 80 MHz. Is a processor SIMD consisting of four mains processing entities: The Global Control Processor (GCP), Digital Input processor (DIP), the Digital Output Processor (DOP) and the Linear Processor Array (LPA).

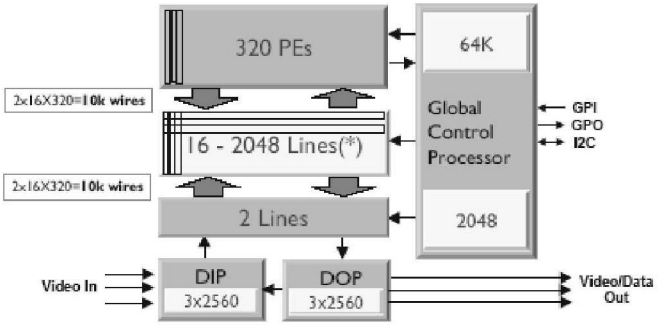

Fig. 3. XeTal Architecture

The GCP is in charge of the program flow operations and the loading of operation into the LPA. The DIP takes care of data handling from the input to the processing memories. The DOP takes the data from the processing memories and send them to the output, the LPA is an array of 320 pixel processors accessing a corresponding part of the processing memory executing simultaneously the same instruction that the GPC controller indicates in the corresponding data stored in the processing memory.

As mentioned before the processing power of the WiCa is derived from the 320 pixel processors encapsulated within the LPA. The Fig. 4.

2.3. Dual Port RAM

DPRAM is a high speed 128K x 9 Dual-Port Static RAM. This device provides two independent ports with separate control address and I/O pins that permit independent, asynchronous access for reads or writes to any location in memory. An

automatic power down feature controlled by the chip permit the on-chip circuitry of each port to enter a very low standby power mode.

The DPRAM plays a very important role in the implementation because given that the data can be accessed randomly, non-linear relation between images or part of images stored in the memory can be implemented and all methods explained in this paper requires such characteristic.

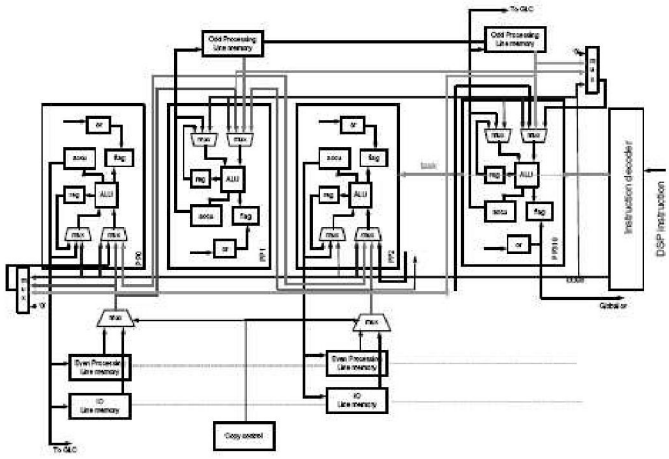

Fig. 4. LPA Structure

3. METHODS

3.1. Non Parametric BGS

In Surveillance it's important to distinguish between what is relevant in the scene and what is not. A very common technique to discriminate between what is static in a scene is Background Subtraction, where a background is represented or modeled and then subtracted from the image to analyze in order to separate the object not belonging to the background.

In real life scenarios we have to deal with constantly changing backgrounds and this characteristic becomes a challenge when a foreground segmentation is needed. Many different methods to address this problem have been proposed, being the most popular Mixture of Gaussian [14] by Stauffer and Grimson, showing a very good accuracy with a relative high cost of memory, but achieving a slow adaptability when new objects are being incorporated in the BG model.

In [8] the same probabilistic base is used when a background is modeled as a bank of images and therefore the velocity of the BG model adaptation is directly proportional of the size of the BG Bank and the refreshing rate in the BG Bank, then for every pixel the probability that the incoming pixel belongs to BG Bank is calculated and a final decision is taken according to (1).

$$Pr(x_t) = \frac{1}{n}\sum_{i=1}^{N} K(x_t - x_i) \tag{1}$$

Where x_1, x_2, ..., x_N are the values of the pixel in the BG bank and Pr is the probability of the X_i pixels belongs into the BG model, and K is the kernel estimator function assumed to be a Normal function $N(0, \Sigma)$.If independence between different colors channels is assumed then ...

$$\Sigma = \begin{pmatrix} \sigma_1^2 & 0 & 0 \\ 0 & \sigma_2^2 & 0 \\ 0 & 0 & \sigma_3^2 \end{pmatrix} \tag{2}$$

Where σ^2 is the variance in the probability function for every channel per every pixel in the BG Model.

We took the work made in [8] and simplified in order to fit in our platform in such way that only one channel is used with BG Bank size of 14 images and a refreshing rate of 1 second.

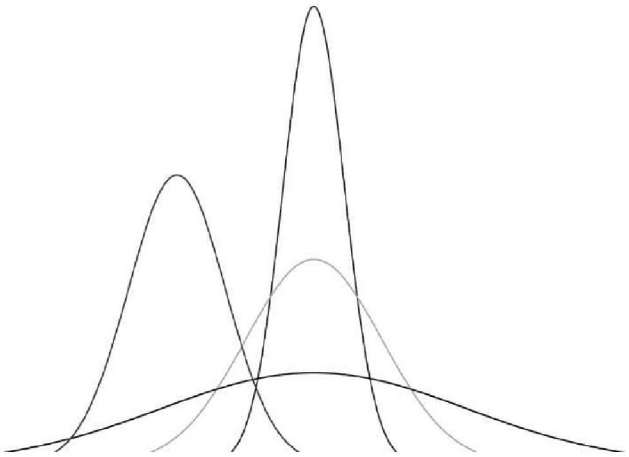

Fig. 5. Probability Density Function for different variances

The variance as seen in the Fig. 5 tells us how spread are the values in the Gaussian distribution, using this variance as a weight the Foreground (FG) can be modeled as the Sum of the absolute difference between the incoming pixel and the mean value of the BG model, multiplying the variance as a weight. The equation (3) shows the FG model and the equations(4) and (5) as intermediate operations.

$$FG = \left[\sum_{i=1}^{N}(x_t - x_i)\right] \cdot \sigma_t^2 > Thr \tag{3}$$

$$\sigma_t^2 = \sum_{i=1}^{N}(x_i - \mu_i)^2 \tag{4}$$

$$\mu_t = \sum_{i=1}^{N} \frac{1}{N} x_i \tag{5}$$

The resulting implementation is functioning in real-time, handling situations of not static backgrounds, like swaying trees or light changes, it's velocity to adapt to new objects merging into the background is very fast (15 seconds). The drawbacks of this implementation are that no shadow suppression is performed since there is used only one color channel, all shadow information is lost.

3.2. Forgetting Foreground (FFG)

In some surveillance scenarios is needed to know when an object has been static for more than certain amount of time. This takes us to analyze the behavior of a pixel in the process between being in the Foreground and see how slowly becomes part of the Background. The Background is modeled as Bank with several pictures taken from the scene as done in Subsection 3.1. When a new pixel alien to the BG model first is tagged as a Foreground because it's value differs drastically from the values in the BG Model. If the pixel in the scene remains the same it's being learned by the BG Model until suddenly is marked as Background.

The Fig. 6 illustrates the ideal case of a pixel corresponding to an object being incorporated in the Background model, as the time passes by, more images on the model have the value of the new object, then the variance in the normal distribution grows reaching a maximum point when half of the background model has the object and half dont have it.

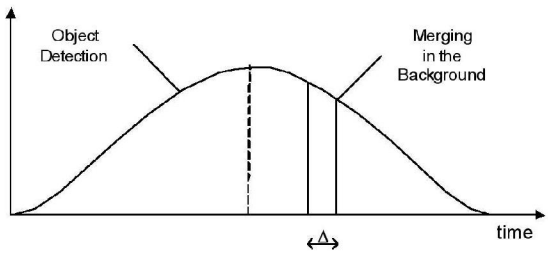

Fig. 6. Ideal Behavior of a pixel incorporating in the BG Model

Then the curve in Fig. 6 can be interpratated as two phases curve, the positive slope telling when a new pixel alien to the BG model is being detected and the negative slope telling when the same pixel is merging into the background model, while the rest of the pixels in the foreground, constantly moving will present a high initial variance but not as well defined as in Fig. 6. Then when we derivate the variance over the time, the first negative derivative of the variance after reaching a maximum variance was achieved will tell us exactly when that pixel is being incorporated into the BG Model. The equation (6).

$$FFG = 1 \quad if \quad \sigma^2_{-1} = max \quad and \quad \sigma^2_{-1} - \sigma^2 > 0 \quad (6)$$

Following this simple concept is possible to catch objects in the transition from foreground to background, and then it can be seen as the foreground is being forgotten by the model.

In Fig. 7 simulations were done with different noises radios, and we can see that even with the noisiest source there is always a well defined curve with no local maximums, and therefore the proposed theory remains true in all noise cases.

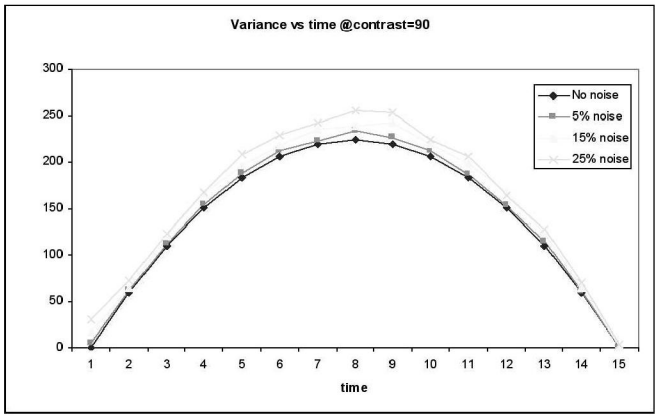

Fig. 7. Simulation of pixel Behavior in the BG Model

Then the questions that pops out is where in the curve we want to catch the object, and the answer would seem very simple, in the negative part of the curve which means that the object is being learned just after the maximum point has been reached.

One of the drawbacks of this approach is that its performance is dependent on the contrast between the background value and the foreground value merging in the background model, since the parameter to analyze is the variance and the variance is an indicator of how different are the values in the BG Model.

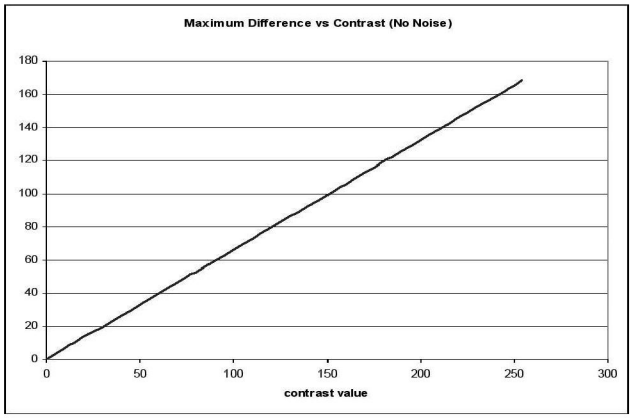

Fig. 8. Maximum Difference vs Contrast

The Fig. 8 shows how the maximum obtained slope increments as the contrast between the two colors increments. In the case of a high difference is chosen as a threshold the system detects correctly the high contrast pixels, but is unable to detect low contrast pixels; if a low difference is chosen, false detections are encounter in high contrast pixels.

3.3. Multidirectional Parallel Image Mapping

The WiCa architecture enables the possibility of accessing randomly an image stored in the memory allowing computing it in a non-linear manner by manipulating the memory address. In this case several copies of the same image are accessed at the same time in order to achieve maximum parallel performance taking advantage of the number of processors working in parallel in the platform architecture.

In this paper the Multidirectional Parallel Image mapping is introduced as an alternative method for labeling and extraction of objects features for the posterior object analysis.

Many labeling methods have been published before, most based on single processors architectures, the method showed in [15] takes color and texture information in order to segment different objects, unfortunately this method is very memory and time consuming and not well suitable in a real time implementation. In [16] and [17] contour tracking methods are slightly faster, yet these methods are meant to be simple processing element based and do not produce a satisfactory performance when the Architecture is SIMD.

In this proposed method we start from a binary image stored in the memory, in order to extract all features of the objects seen in the image, being more specific area and position. The image is mapped in four different directions as seen in Fig. 9, the four images are handled as independent non correlated images. These four images are scanned row by row in the direction shown by the thick line on the right side of the Fig. 9.

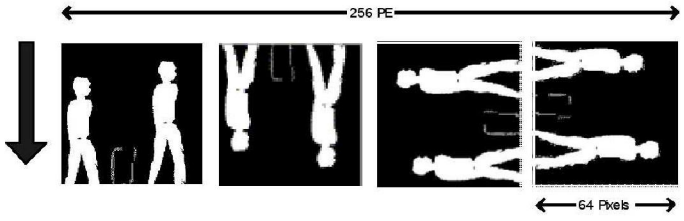

Fig. 9. Multidirectional Parallel Mapping

As the images are being read; horizontal and vertical maximums and minimums are either generated (G) or propagated (P), depending on the connection between the current pixel and the pixels of the previous calculated row. The equations (7), (8) and (9) indicate the logic of how the values are generated. Then four different images are generated containing the four features (Y_{max}, Y_{min}, X_{max}, X_{min}) for every pixel. The four images are written back into the memory for posterior analysis. The Fig. 10 shows the block diagram of how the max and min images are generated and the Area and COG are calculated in a posterior phase.

If the pixel under analysis is empty the value of the max/min is set to 0 as shown in (7) if the pixel is not empty the max/min is generated/propagated with (8).

$$I''(i,j) = 0 \quad if \quad I(i,j) = 0 \quad else \quad I'(i,j) \tag{7}$$

When a non empty pixel is found $I(i,j) \neq 0$, the max/min values are generated or propagated according to (8), where $I'(i,j)$ is the max/min value belonging to each pixel. The functions P() and G() denote Propagation and Generation respectively.

$$\begin{aligned} I'(i,j) &= P\left(I'(i,j-1)\right) \\ &\quad if \quad I'_{j-1} \quad else \quad G(I(i,j)) \end{aligned} \tag{8}$$

The equation (9) is True if at least one of the elements in the previous row is not empty in the range from $-K$ to $+K$. In our case we decided to set K=1.

$$I'_{j-1} = \sum_{j=-K}^{K} I(i,j-1) \geq 1 \tag{9}$$

After having the maximums and minimums per every object in the original image we calculate the bounding box containing every object and the center of gravity (COG). The equations (10) and (11) show the calculations for the Bounding box area and the COG.

$$Area = (X_{max} - X_{min}) \cdot (Y_{max} - Y_{min}) \tag{10}$$

$$COG(x,y) = \left(\frac{X_{max} + X_{min}}{2}, \frac{Y_{max} + Y_{min}}{2} \right) \tag{11}$$

While calculating maximums and minimums four different images are generated containing all connected pixels coordinates. The Fig. 10 shows the block process flow, where the image to be analyzed is scanned in the four directions, the four posterior images are generated and finally the objects features calculations are performed.

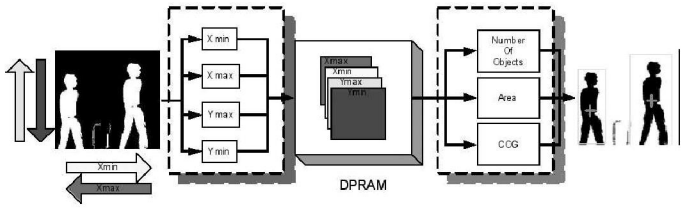

Fig. 10. Multidirectional Parallel Max and Min Calculations

4. EXPERIMENTAL RESULTS

The methods previously presented in this paper have been implemented and satisfactory proved their performance. Improving of the application revealed is still undergoing and therefore all results we are presenting here are experimental and might change at final release.

From the Fig. 11 we can see how the object is being detected as a standing still object after 7 seconds being static, the pixels in white color denote the pixel detected by the method to be FFG pixels. After the object was detected as FFG slowly fades into the background until is no part of the Foreground. We can see how a the chair is split into two different objects in t=10. and finally are no longer detected in t=11.

The FG is shown in Fig. 12 when a person walks and moves within the scene. The application runs real-time at 30 f/s with using 64 Kb of RAM memory, with a minimum working frequency on the XeTal of only 8.18688Mhz in a 640x480 image size. The Fig. 13 shows the DataFlow of the program in the XeTal processor.

5. CONCLUSIONS

In this paper we successfully introduced two new methods that combined with a robust BGS lead to a tracking application implemented in the embedded smart camera WiCa, next generation of this smart camera is pushing hard to overcome some of the issues of the actual one, more resources and the ability to display images in the screen without compromising performance are only a few examples.

There is still work to do to improve this implementation, but it's proved to be effective and efficient with no high computational cost.

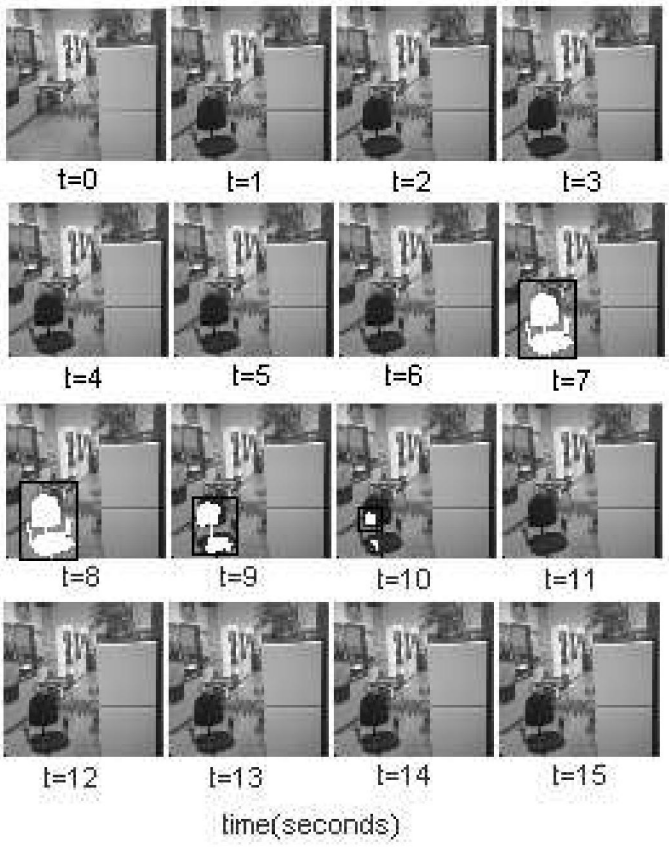

Fig. 11. FFG shows detection after object being static for 7 seconds

6. REFERENCES

[1] R. Collins, "A system for video surveillance and monitoring," *VSAM Final Report Carnegie Mellon Univ. Pittsburg, PA, Tech Rep. CMU-RI-TR-00-12*, May 2000.

[2] P. Meer D. Comaniciu, V. Ramesh, "Real-time tracking of non-rigid objects using mean shift," *IEEE Proc. of CVPR*, vol. 2, pp. 142–149, 2000.

[3] F. Dellaert Z. Khan, T. Balch, "A rao-blackwellized particle filter for eigentracking," *CVPR 04*, pp. 897–900, 2004.

[4] Anamitra Makur R. Venkatesh Babu, "Kernel-based spatial-color modeling for fast moving object tracking," *IEEE ICASSP 07*, vol. 1, pp. 901–904, April 2007.

[5] Anamitra Makur R. Venkatesh Babu, "Robust object tracking with radial basis function networks," *IEEE ICASSP 07*, vol. 1, pp. 937–940, April 2007.

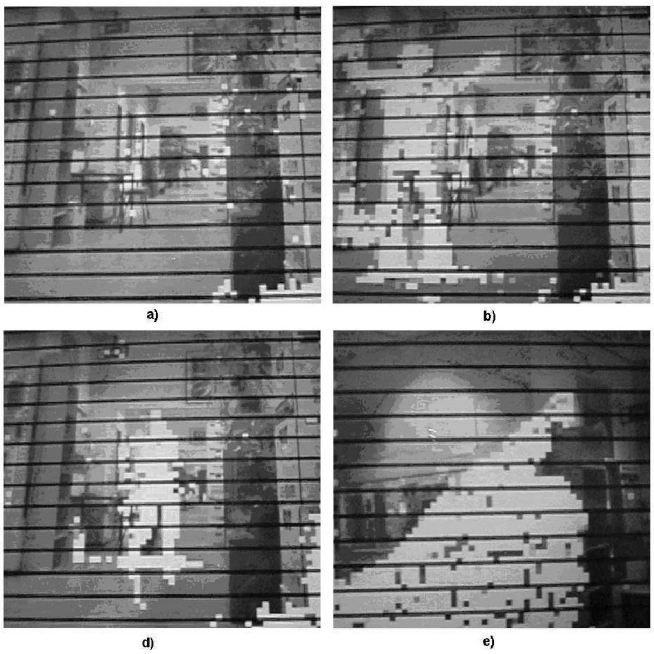

Fig. 12. Objects detected in the Foreground

[6] Stuart C. Schwartz Lan Dong, "Object tracking by finite-state markov process," *IEEE ICASSP 07*, vol. 1, pp. 897–900, April 2007.

[7] Jonathan Connell Max Lu Hans Merkl S. Pankanti Andrew W. Senior Chiao-fe Shu Ying-li Tian Arun. Hampapur, Lisa. Brown, "Multi-scale tracking for smart video surveillance," *IEEE Transactions on Signal Processing*, vol. 22, October 2006.

[8] Larry Davis Ahmed Elgammal, David Harwood, "Non-parametric model for background subtraction," *IEEE ICCV99 Frame-Rate Workshop*, 1999.

[9] A. Danilin Marc Heijligers R. Kleihorst, B. Schueler, "Smart camera mote with high performance vision system," *ACM Workshop on distributed Smart Cameras*, October 2006.

[10] R. Kleihorst B. Schueler E. Simmons, E. Ljung, "Powered distributed wireless smart cameras," *ACM Workshop on distributed Smart Cameras*, October 2006.

[11] R. Kleihorst B. Schueler E. Simmons, E. Ljung, "Distributed vision with multiple uncalibrated cameras," *ACM Workshop on distributed Smart Cameras*, October 2006.

[12] Richard Kleihorst Alexander Danilin Ben Schueler Vincent Jeanne, Francois Jegaden, "Real-time face detection on a dual-sensor smart camera using smooth edges technique," *ACM Conference on Embedded Networked Sensor Systems*, pp. 120–124, 2006.

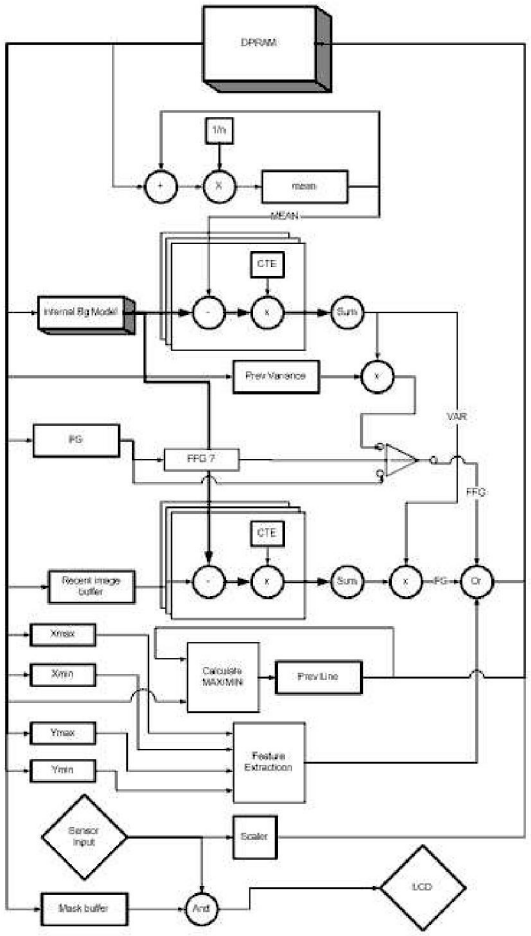

Fig. 13. Program Dataflow

[13] A. van der Avoird M.J.R. Op de Beeck L. Sevat P. Wielage-R. van Veen-H. van Herten R.P. Kleihorst, A.A. Abbo, "Xetal: A low-power high performance smart camera processor," *IEEE ISCAS 2001*, pp. 215–218, 2001.

[14] W. E. L. Stauffer C. Grimson, "Adaptive background mixture models for real-time tracking," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, 1999.

[15] Thomas Leung Jitendra Malik, Serge Belongie and Jianbo Shi, "Contour and texture analysis for image segmentation," *International Journal of Computer Vision*, vol. 43, pp. 7–27, June 2001.

[16] C.J. Lu F. Chang, C.J. Cheng, "A linear-time connected-components labeling algorithms using contour tracing technique," *Journal of CVIU*, vol. 89, pp. 1–23, 2004.

[17] S. Arie W. Kesheng, O. Ekow, "Optimizing connected components labeling algorithms," *Int. Symposium on Medical Imaging*, 2005.