VLSI Implementation of Object-Detection Design using Adaptive Block Partition Decision Algorithm

Ms. S. Devi Poonguzhali,
Assistant professor , Department Of ECE,
K.Ramakrishnan College Of Technology, Tamilnadu ,India.

P. Nancy, M. Swathi, M. Vidhya
U.G Student,Department Of ECE, K.Ramakrishnan College Of Technology, Tamilnadu ,India.

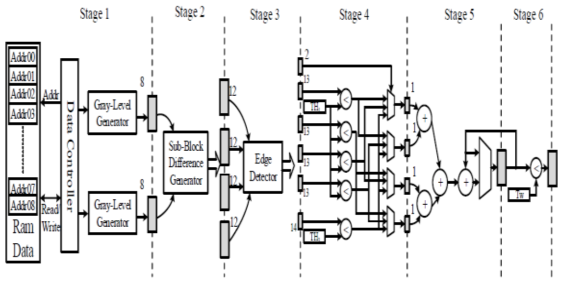
Abstract—In this paper, an adaptive block partition decision methodology is presented for VLSI implementation of object- detection for real-time ultra-high-definition (4K2K) resolution video displaying. The proposed adaptive block partition decision (ABPD) algorithm includes a data controller, a gray-level generator, a sub-block difference module, and an edge detector. The edge detector is designed for discovering edges in images using an efficient edge-catching technique. An adaptive block partition decision technique was added to enhance the shapes of objects and to decrease the edge distortion effects. Furthermore, a threshold constraint is used to set parameters for different sizes of blocks. A statistic methodology of object detection is also used to determine whether it is necessary to trigger an alert signal or not. The VLSI architecture of the proposed design contains 6.99- K gate counts. Its power consumption is 1.63 mW and its operating frequency is to 374.5 MHz by using a 90 nm CMOS technology. Compared with previous designs, the proposed design not only achieves reduction of more silicon area, but also increase the processing throughput, and accuracy of objet- detection for real-time video display.

Index Terms—Application-specific integrated circuits (ASICs), very large-scale integration (VLSI), FPGA, block partition, edge detection, image processing, and object detection .

I.INTRODUCTION

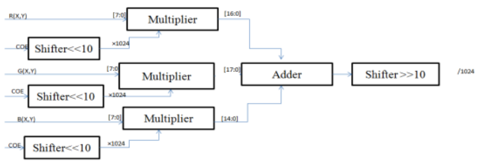
Recently, applications of object detection techniques have become wider and wider. These applications provide powerful solutions for monitoring in real time, screening for analysis and tracking objects or people in the surrounding environment whether in public or private spaces. In common applications, such as high-definition LCD displays [1-2], high- definition video monitoring systems, intelligent surveillance systems [3-4], closed-circuit television (CCTV), smart camera system [5], high-definition road visual surveillance [6], wireless sensor and actual networks (WSAN) [7-8], etc., the usage of the object detection technique is increasing rapidly. As the demand for high-quality video displaying, to develop high-performance 2D or 3D object detecting technologies [9- 10] via the VLSI technique for detecting objects in real time has become a significant trend. Many effective algorithms [11-23] for object-detection have been introduced in related applications. Viola et al. [11] proposed a novel image representation method called Integral Image and an efficient classifier to find Harr-like features for face detection in real time. The split-level colour Haar-like features were used to improve the performance of the object detection in [12]. The frame difference method (FDM) [13] was an effective algorithm to find moving objects by analyzing two nearest frames. Based on the FDM method, a novel application in satellite imaging was proposed in [14]. The FDM uses different times to process a sophisticated analysis in radar images. Although the FDM method takes advantage of high speed and low complexity, the results of the FDM method are focused on the shapes of objects. For this reason, the Gaussian mixture learning model (GMM) was proposed in [15]. The model advanced the performance of object detection by updating the Gaussian distribution according to a learning rate and background image. The background image had ability to update properly according to the Gaussian parameters. A foreground detection algorithm based on a multi-model background maintenance (MBM) methodology was proposed in [16]. It is a GMM-based algorithm and uses additional information from background and foreground images to detect moving objects. Because the FDM-based and GMM-based algorithms are pixel-based detecting techniques, they are suffered noise interference easily and it is difficult to find moving objects when using them. Hence, the block-based background difference subtraction (BBS) [17] was developed for dividing both of current and background images into several blocks for moving objects detection. An illumination- sensitive technique was developed in [18], which detects moving objects by analysing the changing of illumination information. Sermanet et al. [19] proposed an unsupervised algorithm for pedestrian detection. A multiscale convolutional network training method was proposed for scene labeling detection [20]. Saleemi et al. [21] proposed a probability model in visual surveillance for dynamic scene. Due to calculation of visual surveillance is complicated, especially in 3D model-based, a GPU-accelerated methodology was presented in [22]. Moreover, a high-throughput and energy- aware hardware architecture of motion and disparity estimation for searching moving objects was proposed in [23]. Several literatures concerning VLSI architectures for real- time object detection have been presented recently. A memory- efficient VLSI architecture of the GMM algorithm was proposed in [24]. In addition, by using various aspects of underlying segmentation algorithms, this study was successfully implemented via a field-programmable gate array (FPGA). Moreover, the GMM algorithm was also realized by an efficient application-specific integrated circuit (ASIC) [25] and it successfully applied to segment HD video in real time. This study used two different version results to point out that the power, silicon area and throughput are important for VLSI implementation. Cheng et al. [26] used a mathematical operation to analyze scenes for object detection in the fields of the surveillance system. An efficient VLSI design through an AdaBoost detection algorithm was realized in [27]. Kyrkou et al. [28] proposed a FPGA design based on cascade support vector machines (SVM) for real-time classification. A high- throughput FPGA implementation of object detection was proposed in [29], in which a full-image evaluation and fixed- point HOG methodology were used to improve the efficiency of FPGA design. Multi-model background maintenance (MBM) architecture was implemented in a digital circuit [30]. It is worth noting that a general object detection was unsupervised and that the detection result was automatically calculated. In addition, a semi-supervised method was presented in [31], in which a human-machine interaction algorithm was used for VLSI implementation. The results showed that the sophisticated foreground detection can be greatly improved. Javaid et al. [32] proposed a pipelined architecture of multiprocessor system-on-chip (MPSoC) for multimedia applications. The pipelined architecture improved the performance of image processing in real time obviously. Therefore, a block-based background subtraction algorithm and its pipelined architecture was presented in [33]. Although the previous studies [24-33] mentioned above proposed high performance object detection designs, it is required to figure out a real-time and more efficiency object detection hardware- oriented algorithm and its VLSI architecture for real-time video systems. Table I illustrates the basic concepts for previous studies of hardware-oriented object detection algorithms. The process speed of FDM [13-14] and BBS [17] are better than that of GMM [15] and HMIiOL [31]. Although the FDM-based algorithms had benefits of low-complexity and low-memory requirement, the detection quality is worst in these four different bases object detection algorithms. The GMM [15] is an unsupervised algorithm, which detects object by constructing possibility parameters. Hence, the detection quality is better than FDM-based and BBS-based algorithms. However, the GMM cannot endure sudden exchange or objects stay in the foreground image for a long time. In these situations, the GMM parameters will be updated with errors. HMIiOL [31] has the best detection quality in these four different bases algorithms. The semi-supervised technique can set parameters according to user's knowledge. Therefore, it can be used to detect objects in both of static and dynamic scenes. However, the process speed is not fast due to the complexity of HMIiOL algorithm is not efficient enough To be able to achieve objection detecting in real time, the proposed ABPD algorithm was realized by using a VLSI technique. According to the concepts of Table I, the BBS- based method was selected for VLSI implementation due to its fast process speed which also means low complexity. In order to improve the detection quality, an edge detector and an adaptive block partition decision technique were included. In contrast to related works, it had the benefits of low complexity, local detail enhancement, and fewer block filter effects. Moreover, shapes in each block can be adjusted according to the predefined judgement conditions to increase the accuracy of the proposed ABPD algorithm. The proposed object-detection ASIC design achieved the processing of 30 frame- per-second (fps) for 4K2K resolution in the video applications of displaying.

II.ARCHIETECTURE OF ADAPTIVE BLOCK PARTITION DECISION

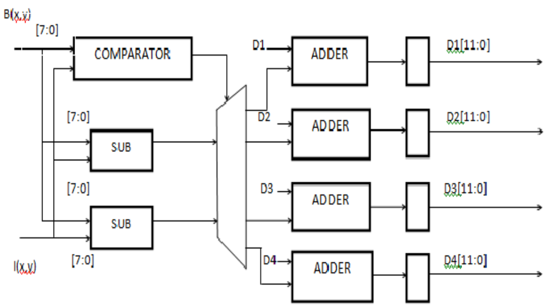


A. Data Controller Since the background and foreground frames are stored in frame memories, a data controller was designed to control the read/write mode and produce addresses to access through the memory device. In addition, the proposed object detection algorithm is based on a regional-based technique, which means that the gray-level generator fetches data in a block format. Hence, the data controller is necessary to achieve the ability to access data with memory according to the order of the blocks. The proposed data controller is an essential design to access the image information which is stored in the memory device.

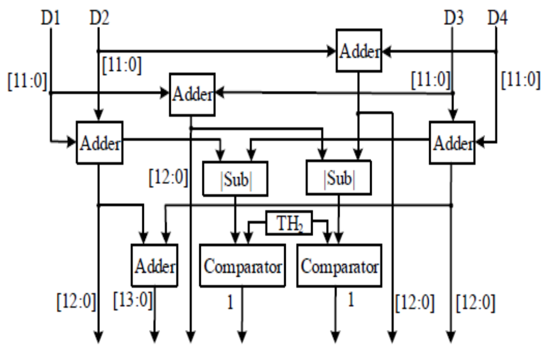
B. Gray-Level Generator: The architecture of the gray-level generator which consists of 4 shifters, 3 multipliers, and 1 adder. The gray-level generator was developed for transferring three eight-bit values of R (x.y), G (x.y), and B (x.y) to one luminance value. In order to decrease the computing complexity of the gray-level coefficients, each coefficient in floating point format was shifted 10-bit left as an integer value. The entire digital gray-level generator provided a low- complexity architecture for the proposed VLSI design.



C. Sub-Block Difference: Generator A hardware-sharing technique was used to design the architecture of the proposed sub-block difference generator to reduce the silicon area. As shown in Fig. 7,the VLSI design of the proposed sub-block difference generator can be realized by a hardware-shared architecture to achieve multiple functions of equations (2), (3), (4), and (5). Because the data were composed of positive integers, the sub-block difference generator consisted of 2 subtractors, 1 comparator, 1multiplexer, and 4 adders. The results of 4 absolute difference values, D1, D2, D3, and D4, can be obtained via the hardware- shared sub-block difference generator design



D. Edge Detector: The cost-efficient and high efficiency edge-detecting method was introduced in ABPD algorithm. The detail information of the edges are be able to obtained by the DH and DV, as shown in equations (7) and (8), respectively.



III . PROPOSED BLOCK DIAGRAM

1) RGB to Gray-Level Transform The Y(x,y) is a luminance value of pixel P(x,y) in the image. It can be calculated by a RGB to luminance equation as $Y(x,y) = 0.299\times R(x,y) + 0.587\times G(x,y) + 0.114\times B(x,y)$ (1) where R(x,y), G(x,y), and B(x,y) are the intensity values of red (R), green (G), and blue (B) colors pixel P(x,y), respectively.

2) Block Difference In order to exploit the advantage of the regional-based technique, a fundamental BBS [17] model was added to the proposed ABPD algorithm. The principle of the proposed ABPD algorithm is shown in Fig. 2. First, each block unit consists of a square, in which the side length of the block is n and divides the block to four sub-blocks (D1, D2, D3, and D4) with the size of n/2×n/2 pixels. Because the BBS algorithm using a block as a unit to detect objects, it can eliminate the discrete noises well. Through this regional-based technique, the precision of the detecting ability will be raised significantly.

Furthermore, the sum of the absolute differences in each sub-block (D1, D2, D3, and D4) is an essential value for the block partition and automatic warning system. Therefore, the sum of the block difference (BD) in each sub-block can be calculated by

BDi(D1) = ∑ ∑ |F(x,y)−B(x,y)| n/2 y=1 n/2 x=1(1)

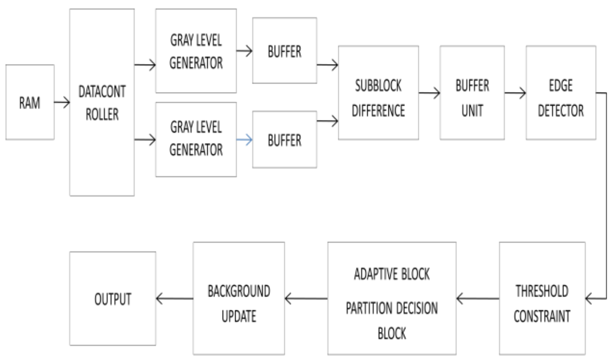
BDi(D1) = ∑ ∑ |F(x,y)−B(x,y)| n/2 y=1 n/2 x=1(2)

BDi(D2) = ∑ ∑ |F(x,y)−B(x,y)| n y=n/2 n/2 x=1(3)

BDi(D3) = ∑ ∑ |F(x,y)−B(x,y)| n/2 y=1 n x=n/2(4)

BDi(D4) = ∑ ∑ |F(x,y)−B(x,y)| n y=n/2 n x=n/2(5)

(i =1, 2, 3…... N N: the total blocks of a frame) where i, F(x,y) and B(x,y) are the indexes of the block number in the input image and the gray-level values of the current frame and background frame, respectively. By calculating the different values between the current frame and background frame, the difference distribution in each sub- block within a block unit will be obtained, which provides important information for the block partition in the following step



3) Threshold Constraint: Because the principle of the proposed ABPD method involves detecting objects by calculating the difference between each block in the background and input images, it is complex to determine whether or not objects are detected for each block. To be able to reduce the complexity of block partition, a threshold constraint is defined to set threshold values according to a threshold constraint curve. The standard deviation is often used to calculate statistical dispersion in probability statics. In this study, the standard deviation is obtained by analyzing the difference between the pixels in foreground and background frames. It reflects the degree of dispersion in the difference frame distribution.

σC = √1 N ∑ (BDi)2 N i=1..... (6)

After calculating the standard deviation (σc) in the specified standard source video, the threshold constraint curve can be obtained as shown in Fig. 3. The threshold value will be re- evaluated once the length of side in each block or standard deviation are changed. In Gaussian distribution, over 99.5% of data are distributed in the range of four times standard deviations. By using this characteristic, the difference value which is compared by the foreground and background exceed four times standard deviations will be considered as objects. If the difference value in the range of four times deviations, it will be regarded as normal deviation, which is background. Therefore, four times standard deviations and the square of the side length were used to satisfy the entire block threshold TH3 in this study. Furthermore, half of whole block threshold TH1 was used in the divided block. In order to switch between TH3 and TH1 properly, a threshold TH2, the average value of TH3 and TH1, was provided. The block partition technique can be implemented by using the proposed low-complexity threshold constrain.

4) Block Partition : Although separating a frame into many blocks can promote the accuracy of the object detection, the edge information will be missed due to the blocking effect. For example, if a block in an image includes edge information, an abruptly changing brightness would interfere with the results of object detection. Such situations will cause some unnatural artefacts to appear in the results. Hence, it is needed to develop an edge detection technique to enhance the object detection ability when the blocks are divided into numerous sub-blocks. For this reason, the edge information within a block is very important. Hence, the first step of the block partition procedure is to find the continuous features within a block, which is helpful for segmenting the block for object detection. Fig. 4 shows three types of the block partition. Because the proposed algorithm is created for VLSI implementation, it is necessary to develop a low-complexity methodology for discovering the edge information. Hence, the continuous features can be obtained by analyzing the difference in the values in the horizontal and vertical directions.

The difference in the horizontal direction (DH), which is around the boundary of the blocks, can be evaluated by

DHi = |(BDi(D1) +BDi(D2))−(BDi(D3) +BDi(D4))| (7)

The difference in the vertical direction (DV) which is around the boundary of the blocks can be evaluated by DV i = |(BDi(D1) +BDi(D3))−(BDi(D2) +BDi(D4))| (8)

To obtain the edge information related to the horizontal (DH) and vertical (DV) directions, some threshold values were used for judgement. For example, while the DH value is greater than that of TH2, and the DV value is less than TH2 value, the result of the block difference (BR) is used to strengthen the edges of object in the horizontal direction by

BRi(k1) = BRi(k2) = {1,BDi(D1)+BDi(D2)> TH1 0, otherwise (9)

BRi(k3) = BRi(k4) = { 1,BDi(D3)+BDi(D4)> TH1 0, otherwise (10)

where TH1 is a threshold value that is compared with two sub- blocks. The k1, k2, k3, and k4 represent the binary results of the block difference. Otherwise, while the DV value is greater than the TH2 value, and the DH value is less than TH2 value, the result of block difference (BR) is used to strengthen the edges of the objects in the vertical direction by

BRi(k1) = BRi(k3) = { 1,BDi(D1)+BDi(D3)> TH1 0, otherwise (11)

BRi(k2) = BRi(k4) = { 1,BDi(D2)+BDi(D4)> TH1 0, otherwise (12)

Also, while both of the values of the DV and DH are more or less than that of TH2, the result of the block difference (BR) is calculated without any edge enhancement by

BRi(k1) = BRi(k2) = { 1 , BDi(SUM) > TH3 0, otherwise (13) BRi(k3) = BRi(k4) = { 1 , BDi(SUM) > TH3 0, otherwise (14)

where SUM represents the sum of the absolute differences in four sub-blocks (D1+D2+D3+D4) and the TH3 is the specific value of the threshold values for the four sub-blocks. Finally, each of the block differences can be required by the detection result of 1 or 0 according to the comparison procedure

B. Automatic Warning System

1) Statistics Different Sub-Blocks: When the detection results of the block difference (BR) in the block partition step is finished, a total difference (TD) value can be obtained by counting the block difference (BR) values with the parameters k1, k2, k3, and k4 by

TD = ∑ (BRi(k1) N i=1 +BRi(k2)+BRi(k3) +BRi(k4)) (15)

where TD is the total amount of sub-block difference values, and N is the number of blocks in the detected results. The value of TD is determined whether object is detected or not

2) Alert Signal: An alert signal (AS) is generated conditionally in the proposed automatic warning system. It is helpful to express directly the detection results. Therefore, a threshold value TW was developed for judging whether any object was detected in a critical region or not. Based on the calculation of the proposed threshold constraint curve, the weight of TW can be decided by 0.5% of pixels in a frame of the source video. The alert signal (AS) can be obtained by

AS = { 1, TD > TW 0, otherwise (16)

According to the relation of TD and TW, the alert signal will be generated and produced for the proposed automatic warning system

3) Background Update : Because the proposed ABPD algorithm is based on the regional-based technique, a noiseless background image is necessary to be constructed. A cumulative value was used in this study to decide whether or not update the background image with a new frame. By adding the proposed background update technique, the background image can be updated properly. Through the background update procedure flow mentioned above, the accuracy of the proposed ABPD algorithm can be improved greatly

IV.RESULT

The FP rate can be obtained by:

FP Rate= fp fp+tn (17)

The recall can be calculated by: Recall = tp tp+fn (18)

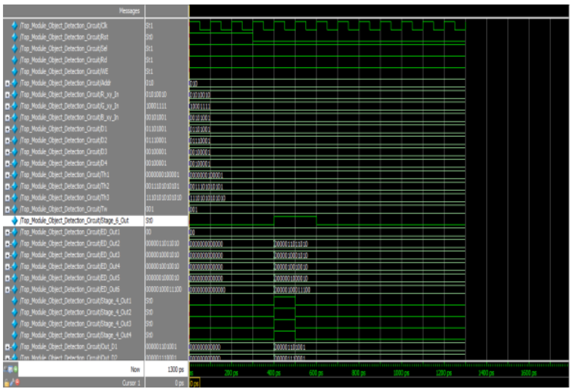
The precision can be measured as: Precision = tp tp+fp (19)

The value of similarity can be defined as: Similarity = tp tp+fn+fp (20)

where fp, fn, tp, and tn represent "false positive", "false negative", "true positive" and "true negative", respectively. F-measure [35] which is another parameter to be used to analyze the performance of the detected results by various algorithms. F-measure is the object of coverage area between detect results and ground truth, it means the tracking degree of ground truth and estimation. The F-measure can be measured as:

F-measure = 2× ($precision$×$recall$) ($precision$+$recall$) (21)

In order to calculate the standard deviation parameters (σc) for the proposed ABPD algorithm, a background maintenance database [36] was selected as the training video data set. After training the parameters by the background maintenance database [36], the standard source videos obtained from Purdue University [37] and benchmark database [38] were used to analyze the performance of the proposed ABPD algorithms as shown in Fig. 9, in which three testing standard patterns were obtained from Purdue University [37] in the United States of American (USA).The duration length in the testing patterns of AESOS_01, AESOS_02, and AESOS_03 are 44, 44, and 50 seconds, individually. The frame size is 480 ×640 pixels and the frame numbers are 1341, 1341, and 1501, individually.



V.CONCLUSION

In this study, a novel low-complexity, edge-catching and block-partition hardware-oriented object-detection algorithm is presented. Through the division of blocks into four sub-blocks, the edge distortion was reduced efficiently. As simulation results show, the proposed ABPD algorithm achieved higher adaptability and object similarity than did the previous low-complexity studies. It helped to successfully develop a low-cost and high-speed object-detection VLSI implementation for real-time 4K2K video applications. Compared with previous designs, the proposed design provided higher accuracy, lower cost and better capability than did the previous low-complexity object detection designs.

REFERENCES

[1] Honghai Liu, Shengyong Chen, and Naoyuki Kubota, "Intelligent video systems and analytics: a survey," IEEE Transactions on. Ind. Inform., vol. 9, no. 3, pp. 1222-1233, April 2013.
[2] J. S. Kim, D. H. Yeom, and Y. H. Joo, "Fast and robust algorithm of tracking multiple moving objects for intelligent video surveillance systems," IEEE Trans. Consumer Electronics, vol. 57, no. 3, pp. 1165-1170, Aug. 2011.
[3] Carlos C. and Narciso G., "Efficient moving object detection for lightweight applications on smart cameras," IEEE Transactions on Cir. and Sys. for Video Tech., vol. 23, no. 1, pp. 1-14, June 2013.
[4] H. Sheng, C. Li, Q. Wen, and Z. Xiong, "Real-time anti-interference location of vehicle license plates using high-definition video," IEEE Intell. Transp. Syst. Mag., vol.1, no.4, pp.17-23, winter 2009
[5] X. Cao, J. Chen, Y. Xiao, and Y. Sun, " Building environment control with wireless sensor and actuator networks: Centralized versus distributed," IEEE Trans. Ind. Electron., vol.27, no.11, pp.3596-3605, Nov. 2010.
[6] D. Wu, S. Ci, H. Luo, Y. Ye, H. Wang, "Video surveillance over wireless sensor and actuator networks Using Active

Cameras", IEEE Trans. on Automatic Control, vol.56, no. 10, pp 2467-2472, Aug. 2011.

[7] M. Emoto, Y. Kusakabe, and M. Sugawara, "High-Frame-Rate motion picture quality and its independence of viewing distance", J. Display Technol., vol. 10, no. 8, pp 635-641, Aug. 2014.

[8] M. C. Kim, "Fourier-Domain analysis of display pixel structure for image quality", Journal of Display Technology, vol. 12, no. 2, pp 185-194, Feb. 2016.

[9] H. N. Fructuoso, M. Martinez-Corral, G. S. Tortosa, A. P. Marti, and B. Javidi, "Photoelastic analysis of partially occluded objects with an integral-imaging polariscope", J. Display Technol., vol. 10, no. 4, pp 255-262, Apr. 2014.

[10] H. Unno, S. Isaka, Y. Takashima, and K. U., "New technique for embedding depth information in captured images using light beam containing invisible high-frequency patterns-design and preparation of new experimental setup with some comments", Journal of Display Technology., vol. 11, no. 2, pp 136-145, Feb. 2015.

[11] Paul Viola, and Michael J. Jones, "Robust real-time face detection," International Journal of Computer Vision, pp. 137-154, 2004. [12] C. Liu, Faliang Chang, and Chengyun Liu, "Cascaded split-level colour Haar-like features for object detection," Electronic Letters, vol. 51, no. 25, pp. 2106-2107, Dec. 2015.

[12] S. Y. Chien, S. Y. Ma, and L. G. Chen, "Efficient moving object segmentation algorithm using background registration technique," IEEE Trans. Circuits Syst. Video Tech., vol. 12, no. 7, pp.577–586, July. 2002.

[13] T. Celik, "Multiscale change detection in multitemporal satellite images," IEEE Geoscience and Remote Sensing Letters, vol. 6, no. 4, pp. 820-824, Aug. 2009.

[14] Dar-Shyang Lee, "Effective Gaussian mixture learning for video background subtraction," IEEE Transactions on Pattern Analysis and Machine Intell., vol. 27, no. 5, pp. 827-832, May 2005.