Efficient Adaptive Background Subtraction Based on Multi-resolution Background Modelling and Updating

Ruijiang Luo[1], Liyuan Li[1], and Irene Yu-Hua Gu[2]

[1] Institute for Infocomm Research, Singapore
{rjluo,lyli}@i2r.a-star.edu.sg
[2] Dept. of Signals and Systems, Chalmers Univ. of Technology, Sweden
irenegu@chalmers.se

Abstract. Adaptive background subtraction (ABS) is a fundamental step for foreground object detection in many real-time video surveillance systems. In many ABS methods, a pixel-based statistical model is used for the background and each pixel is updated online to adapt to various background changes. As a result, heavy computation and memory consumption are required. In this paper, we propose an efficient methodology for implementation of ABS algorithms based on multi-resolution background modelling and sequential sampling for updating background. Experiments and quantitative evaluation are conducted on two open data sets (PETS2001 and PETS2006) and scenarios captured in some public places, and some results are included. Our results have shown that the proposed method requires a significant reduction in memory and CPU usage, meanwhile maintaining a similar foreground segmentation performance as compared with the corresponding single resolution methods.

Keywords: Adaptive background subtraction, multi-resolution modelling, principal feature representation, statistical modelling.

1 Introduction

Adaptive background subtraction (ABS) is a fundamental step in video surveillance [1,2,3,4]. A video surveillance system often employs a stationary camera directing at the scene of interest. A background model is then generated and dynamically maintained to follow the background changes.

Much work has been done on adaptive background subtraction (ABS) using pixel-based statistical modelling. Wren [3] employed a single Gaussian model to describe the color distribution of each pixel. In [4], a model of mixture of Gaussians (MoG) is proposed to handle more complicated situations, e.g., moving bush under windy conditions. Many enhanced variants of MoG have been proposed. Some integrated the gradients [5], depth [6], or local features [7] into the Gaussians. Others employed the non-parametric models, e.g. kernels, to replace the Gaussians [8,9]. In [10], a model of principal feature representation (PFR)

H.H.S. Ip et al. (Eds.): PCM 2007, LNCS 4810, pp. 118–127, 2007.

was proposed to characterize each background pixel. Using PFR, multiple features from the background, such as color, gradient, and color co-occurrence, can be learned online and used for classification of background and foreground.

By employing various statistical models and multiple features for background modelling, adaptive background subtraction (ABS) methods become robust with respect to a variety of complex backgrounds. The price, however, is the requirement of large memory space and heavy computation [11]. This makes the methods difficult to be applied to real time surveillance on high-resolution images.

It is observed that for images captured by surveillance cameras in public places, most pixels belong to some objects or patches, e.g., road surfaces, vegetation and sky. Such pixels only contain small local feature variations. It indicates that a single statistical model can be employed to monitor a local patch in such smooth image regions. For those small percentage of image pixels that are associated with neighborhoods containing high local visual feature variations, *e.g.* edges between smooth regions, individual statistics is required to accurately characterize each pixel. Motivated by the above, we propose a novel method of multi-resolution adaptive background subtraction (MRABS) for efficient foreground detection. Compared to the region-based method in [12], ours uses gradient statistics to select smooth patches with a fixed memory consumption for background modelling, which is more robust for long-term running and easier for hardware implementation. Meanwhile, a sequential sampling is proposed to improve the efficiency of model updating.

The proposed method is implemented on both PFR-based and MoG-based algorithms. Our analysis shows that using the proposed method, only around 1/8 of memory and 1/6.4 of CPU resource are needed. In real implementation, some extra memory and computations are required. Overall, for a similar background subtraction performance, it is found that the multi-resolution PFR-based algorithm requires about 20.7% memory space and 29.4% CPU consumption as compared with the single-resolution version of the algorithm, while the required memory space and CPU usage for the multi-resolution MoG-based algorithm are reduced to 36.5% and 57.5% as compared with its single-resolution version.

The rest of the paper is organized as follows. Section 2 describes the multi-resolution modelling method, including the analysis of computational efficiency. Section 3 describes the experiments with some results and performance evaluation included. Finally, conclusions are given in Section 4.

2 Multi-resolution Adaptive Background Subtraction

The proposed method contains four parts: Multi-Resolution (MR) Management, MR Background Modelling, MR Background Subtraction, and MR Model Updating, as shown in Fig.1. To make it easy to understand, we use PFR-based algorithm as the example. However, the proposed multi-resolution background maintenance method can also be applied to other algorithms in a similar manner, e.g. we have applied the method to the MoG-based algorithm.

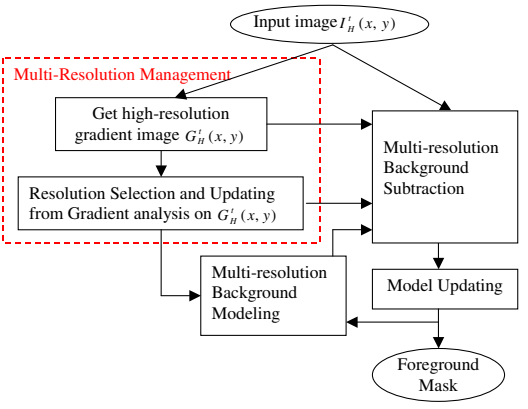

Fig. 1. Block diagram of the multi-resolution adaptive background subtraction method

2.1 Multi-resolution Management

To achieve multi-resolution background modelling, a high resolution image is first divided into small blocks of fixed size ($W_B \times H_B$ pixels). A block is classified as either low or high resolution based on the statistics of local variations.

Since image gradient is a good feature to indicate local variations, we use an accumulated gradient feature, *variance of the gradient power*, for resolution management. Let (g_x,g_y) be the gradient vector generated by a Sobel operator at the pixel $\mathbf{x}=(x,y)$ in frame I_t. The power of gradient, $G_H^t(\mathbf{x}) = g_x^2 + g_y^2$, is then accumulated along time using

$$\tilde{G}_H^t(\mathbf{x}) = \alpha \cdot G_H^t(\mathbf{x}) + (1-\alpha)\tilde{G}_H^{t-1}(\mathbf{x}) \tag{1}$$

where α is a constant used as a smooth factor (α=0.01 in our tests). The variance of the gradient power for the i-th block is computed over all pixels in the block,

$$\sigma_{t,i}^2 = E(\tilde{G}_H^t - E(\tilde{G}_H^t))^2 \tag{2}$$

where $E(\cdot)$ is the expectation. Since most blocks have smooth local neighborhoods, the corresponding variances $\sigma_{t,i}^2$ are small. From the histogram of $\sigma_{t,i}^2$ over all blocks in the image, a small threshold value Th can be found such that the histogram area below this threshold covers γ_m% of image blocks. These blocks are set as the low resolution blocks. The $1-\gamma_m$% blocks that exceed the threshold (i.e., having high local variations) are assigned as high resolution blocks. Examples of multi-resolution block representation on several scenes are shown in Fig.2. With a fixed γ_m, the memory usage is also fixed.

All blocks are initially set as low resolution when the system starts. The resolution of each block is then updated every t_{train} seconds by the resolution management module: For the i-th block, if $\sigma_{t,i}^2 \geq Th$ is satisfied at time t and block was in low-resolution at $t-1$, it is changed to a high resolution block. Conversely, if the i-th block was in high resolution and $\sigma_{t,i}^2 < Th$ is satisfied at time t, the block is switched to low resolution block.

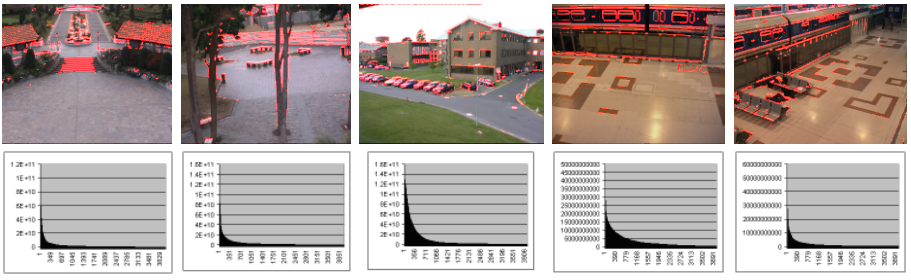

Fig. 2. Images contain high/low resolution blocks and their variance of gradient power values. Row-1: images where small red rectangles denote high-resolution blocks, $\gamma_m = 90$. Row-2: the corresponding sorted histograms of $\sigma^2_{t,i}$ (showing the top 16%)

2.2 Multi-resolution Background Modelling

Since the low-resolution blocks represent areas with low local variations, it implies that pixels within these blocks have similar colors and gradients. For PFR-based method, the features of co-occurrences (M_{cc}), which are employed as the feature for dynamic background, can be deleted. For a high-resolution block, all three types of the principal features (M_c, M_e, M_{cc}) are maintained at each pixel in the block. The number of these principal features is described in Table 1.

Table 1. Values of parameters

Parameter	Value
M_c: number of principle colors	30
M_e: number of principle gradients	30
M_{cc}: number of color co-occurrence	60
(W_B, H_B): width and height of each block	$(4, 4)$

Let B_i be a high-resolution block, and $N_B = W_B \times H_B$ be the size of the block (4×4 in our tests). The tables for the PFR algorithm can be expressed as

$$T_v(B_i) = \{T_v^i(\mathbf{x}_j)\}_{j=1}^{N_B} \tag{3}$$

for the j-th pixel $\mathbf{x}_j \in B_i$, its feature vector contains 3 component vectors: color, gradient and color co-occurrence ($\mathbf{v} = \mathbf{c}, \mathbf{e}$ and $\mathbf{cc}$). For each component feature vector, the table can be expressed by

$$T_v(\mathbf{x}_i) = \{p_v^{i,t}(b), \{S_v^{i,t}(l) = (p_{v_l}^t, p_{v_l|b}^t, v_l)\}_{l=1}^{M_v}\} \tag{4}$$

All together, $3 \times N_B$ tables are used for each block. We use unsigned char (1 byte) for color vector $\mathbf{c}$ and color co-occurrence vector $\mathbf{cc}$, short integer (2 bytes) for gradient vector $\mathbf{e}$ and floating point (4 bytes) for all the possibilities p. The size of the features are shown in Table 1. We can estimate the memory space required for the three principal features at a pixel in a high resolution block by:

$$\begin{aligned} m_e &= (2S_i + 2S_f) \times M_e + S_f = 364 \; bytes \\ m_c &= (3S_c + 2S_f) \times M_c + S_f = 334 \; bytes \\ m_{cc} &= (6S_c + 2S_f) \times M_c c + S_f = 844 \; bytes \end{aligned} \tag{5}$$

If the block B_i is assigned as a low-resolution block, then there is one table for principal colors and one table for principal gradients in the block:

$$\begin{aligned} T_c(B_i) &= \{p_c^{i,t}(b), \{S_c^{i,t}(l) = (p_{c_l}^t, p_{c_l|b}^t, c_l)\}_{l=1}^{M_c}\} \\ T_e(B_i) &= \{p_e^{i,t}(b), \{S_e^{i,t}(l) = (p_{e_l}^t, p_{e_l|b}^t, e_l)\}_{l=1}^{M_e}\} \end{aligned} \tag{6}$$

Based on this, we can compute the storage space for different types of block, which is $m_{lb} = m_e + m_c = 698$ *bytes* for a low resolution block and $m_{hb} = (W_B \times H_B) \times (m_e + m_c + m_{cc}) = 24672$ *bytes* for a high resolution block.

Let $N_I = M \times N$ be the image size, $\gamma_m\%$ be proportion of the low-resolution blocks, N_{hb} and N_{lb} be the number of high-resolution and low-resolution blocks, respectively, where

$$\begin{aligned} N_{hb} &= \tfrac{N_I}{W_B \times H_B} \times (1 - \gamma_m\%), \\ N_{lb} &= \tfrac{N_I}{W_B \times H_B} \times \gamma_m\% \end{aligned} \tag{7}$$

Set $\gamma_m\% = 90\%$, from Table 1 one can obtain $N_{hb} = 0.00625 N_I$ and $N_{lb} = 0.05625 N_I$. Hence, the total memory consumption for the background in the multi-resolution PFR-based models is $mem_{MR} = m_{lb} \times N_{lb} + m_{hb} \times N_{hb} \sim 193 N_I$. Original single resolution PFR method equivalents to treating all blocks as in high resolution, the required memory space is $mem_{normal} = 1542 N_I$. Hence, the required memory space of the multi-resolution PFR-based method is reduced to

$$mem_{MR} \sim \frac{193 N_I}{1542 N_I} \sim \frac{1}{8} mem_{normal} \tag{8}$$

2.3 Multi-resolution Background Subtraction

Under multi-resolution background modelling, the background model of the block is used for background and foreground classification if a pixel is in a low-resolution block. If a pixel is in a high-resolution block, the background model of that pixel is used. Since the computational in feature matching is high for the PFR-based method, the following coarse to fine process is proposed.

First, *background differencing* (BD) between input frame and the maintained background image, and *temporal differencing* (TD) between two consecutive input frames are performed at a lower resolution. The results are then zoomed-in to the original resolution to yield an initial coarse foreground mask. In most scenarios captured by a surveillance camera, a large portion of the image does not contain foreground objects. As a result, much CPU power can be saved. To keep small objects of interest in scene, a quarter-sized image is used for the BD and TD operations (i.e., $W_L = (1/2)W$, $H_L = (1/2)H$).

Next, Bayesian classification is performed pixel by pixel on the obtained foreground mask to refine the segmentation. For a pixel in a low-resolution block, only color and gradient are used. For a pixel in a high-resolution block, all three features, color, gradient, and color co-occurrence, are taken into consideration.

2.4 Multi-resolution Background Maintenance

Different updating strategies are applied to the blocks of different resolutions. For a high-resolution block, pixel by pixel updating operation is applied. However, for a low-resolution block, the following sequential sampling method is proposed since the visual features from different pixels inside the block are similar and stable through the time: at each time step, the background model of the block is updated by the features from one pixel sequentially sampled from the block, as indicated by Fig.3.

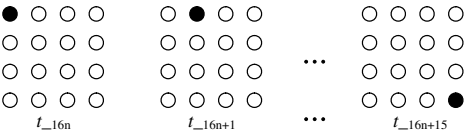

Fig. 3. Updating the background model for a 4×4 block using the sequential sampling

Using 4×4 blocks, a pixel in a low-resolution block is sampled once every 16 frames to update the block background model. Similarly, we can obtain the computational cost for the updating with respective to conventional method,

$$Update_{MR} = \frac{N_{lb}+(W_B \times H_B \times N_{hb})}{N_I} Update_{normal} = \frac{1}{6.4} \times Update_{normal} \quad (9)$$

That implies that only 15.6% of the updating time is needed as compared to conventional single resolution updating routine.

3 Experimental Results

The proposed method has been tested on image sequences from several open data sets, including PETS 2001 and 2006 data sets, and some sequences captured in the public places at Santosa (Singapore). In our tests, 10% of the blocks are set as high resolution (i.e., $\gamma_m\% = 90\%$). Our tests were conducted using both multi-resolution PFR-based method and multi-resolution MoG-based method.

3.1 Evaluation: Computational Cost and Memory Usage

The text results in Table 2 shows the average memory consumption and frame rate of PFR-based and MoG-based background subtraction operations on the PETS data set with the conventional single resolution and the proposed multi-resolution technique. The results were obtained using a 3.0GHz Dell Desktop with 1GB memory. In the real implementation, some extra memories are needed to save temporal results. Therefore, the obtained memory usage is higher than the theoretical analysis.

Table 2. Average memory consumption for the PETS dataset (image size 768×576): the conventional single resolution (SR) technique vs. the proposed multi-resolution (MR) technique

Method	Using SR	Using MR
Principal Feature Representation (PFR)	870MB, 1.62fps	180MB, 5.5fps
Mixture of Gaussians (MoG)	63MB, 6.3fps	23MB, 11.0fps

One can observe that by employing multi-resolution strategy, the processing speed of the PFR-based algorithm is increased by 3.4 times, with only about 20.7% of the memory space consumption as compared with the conventional single resolution method. That is, to reach real-time processing ($\geq 8fps$) using PFR-based algorithm, previously one system can only process one input color stream at a small resolution of 176×144. With the proposed multi-resolution technique, the same system can now process two input color streams of 352×288 resolution at 11fps each without any hardware upgrading. For some cases where a lower frame rate is acceptable, the reduced memory requirement enables one system to process even more inputs at same time.

For the multi-resolution MoG-based method, we achieved less significant improvements, with nearly doubling the processing speed, and requiring only 36.5% of memory as compared with conventional single resolution alternative. It is because the feature matching in MoG algorithm is really simple. It actually takes less time to perform direct feature matching for foreground and background classification on image in original resolution than the coarse to fine operations (image zooming down, "TD", "BD", and zooming back to its original size). But for mass deployment, this 50% resource saving could be rather significant.

Table 3. Detailed processing time for modules: SR technique vs. MR technique

	TD	BD	GD	Classification	Updating	Others
PFR	4.55	7.82	0.54	5.79	37.25	4.36
MR-PFR	1.08	2.07	0.33	3.20	9.19	2.24
MoG	-	-	-	6.87	9.33	-
MR-MoG	-	-	0.39	4.80	2.94	0.94

Table 3 shows some details on how much time each module takes in PFR-based and MoG-based background subtraction operations with and without the proposed multi-resolution technique. Each item is average time consumed (in seconds) for processing 100 input frames of 768×576 resolution. "TD" and "BD" represent temporal differencing and background differencing, respectively. "GD" means gradient detection using Sobel algorithm, "Classification" is foreground and background classification, "Updating" means background model updating, and "Others" is for all other processing, such as memory copying to save temporary results, etc. For original PFR-based method, the "TD" and "BD" operations are performed on the input resolution images, while those in MR-PFR method are performed on quarter-size inputs. It clearly shows the proposed MR technique can well improve the efficiency of both complicated (e.g. PFR) and simple (e.g. MoG) background subtraction algorithms.

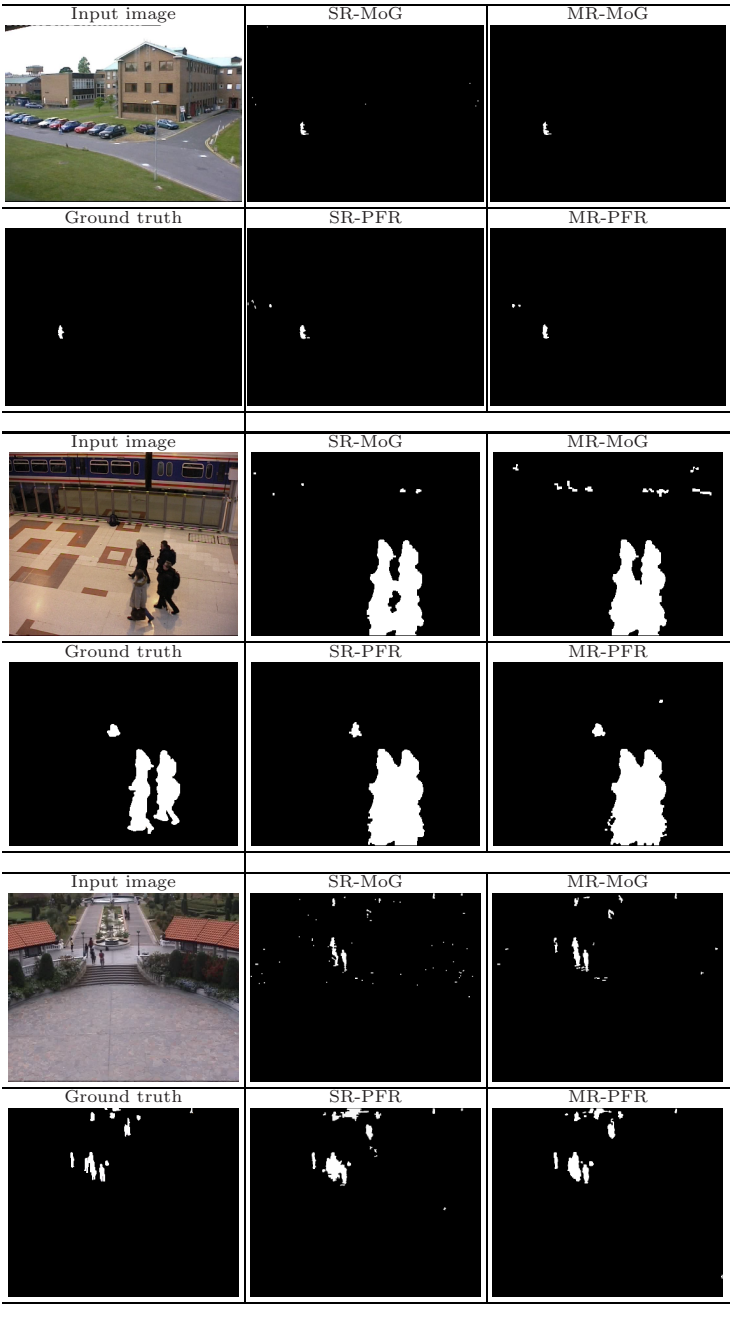

Fig. 4. Background subtraction: conventional single resolution (SR) method vs. the proposed multi-resolution (MR) technique. Rows 1 to 2: PETS2001 Dataset-1 Camera-1; Rows 3 to 4: PETS2006 Scene-3 Clip-1; and Rows 5 to 6: Santosa Dataset-1

3.2 Evaluation: Effectiveness of the Method

The performance of the foreground segmentation by adaptive background subtraction is evaluated and compared for algorithms using the conventional single resolution and the proposed multi-resolution technique . The quantitative evaluation is performed on three blindly picked sequences from the testing data set, they are: PETS2001 Dataset-1 Camera-1, PETS2006 Scene-3 Clip-1 and Santosa Dataset-1.

For each sequence, processing results are sampled on every 100-frame intervals. The segmentation results of these sample frames are then compared with the manually generated "ground truths". The example of the segmented results from the PFR-based and MoG-based methods with single and multi- resolution, and the ground truth are shown in Fig.4.

To further evaluate the method, we use the metric defined as the ratio between the intersection and the union of the ground truth and the segmented regions, as used in [10],

$$S(A, B) = \frac{A \bigcap B}{A \bigcup B} \tag{10}$$

Table 4 includes the resulting metric values for the two methods, PFR and MoG, with and without applying multi-resolution technique. According to [10], the performance is rather good if $S > 0.5$ and is nearly perfect if $S > 0.8$. Since the regional information from all pixels is used to update its background model along the time in the low-resolution blocks, and most blocks belong to low-resolution, from Table 4, it is observed that one can significantly improve the system efficiency with very little sacrifice of the effectiveness by using the proposed multi-resolution technique. For most cases on the PETS dataset, where the images have higher quality, the system performance are even slightly improved.

Table 4. The resulting metric values S (defined in Eq.(10)) for quantitative evaluation and comparison of the effectiveness of adaptive background subtraction methods: single resolution (SR) technique vs. multi-resolut (MR) technique

Name of dataset	SR-PFR	MR-PFR	SR-MoG	MR-MoG
PETS01 Dataset-1 Camera-1	0.7	0.8	0.6	0.65
PETS06 Scene-3 Clip-1	0.74	0.76	0.51	0.6
Santosa	0.75	0.73	0.5	0.47
Average	0.73	0.763	0.537	0.573

4 Conclusion

The proposed multi-resolution background maintenance method, aimed at improving the efficiency on memory usage and computational cost in adaptive background subtraction, has been applied and tested to the principal feature representation (PFR)- and the mixture of Gaussians (MoG)-based methods. By dividing each input image into fix-size high and low resolution blocks using a gradient-based analysis, and using a sequential sampling method for updating the background model, we have achieved 3.4 times faster speed in computation,

with only 20.7% memory consumption as compared with the conventional pixel-based PFR algorithm. For MoG-based method, the proposed multi-resolution approach has resulted in 1.74 times faster speed, and requires 36.5% of memory space as compared with its pixel-based correspondence.

References

1. Haritaoglu, I., Harwood, D., Davis, L.: W^4: Real-time surveillance of people and their activities. IEEE Trans. PAMI 22(8), 809–830 (2000)
2. Hu, W., Tan, T., Wang, L., Maybank, S.: A survey on visual surveillance of object motion and behaviors. IEEE Trans. Systems, Man, and Cybernetics, Part C 34(3), 334–352 (2004)
3. Wren, C., Azarbaygaui, A., Darrell, T., Pentland, A.: Pfinder: Real-time tracking of the human body. IEEE Trans. PAMI 19(7), 780–785 (1997)
4. Stauffer, C., Grimson, W.: Learning patterns of activity using real-time tracking. IEEE Trans. PAMI 22(8), 747–757 (2000)
5. Javed, O., Shafique, K., Shah, M.: A hierarchical approach to robust background subtraction using color and gradient information. In: Proc. IEEE Workshop Motion and Video Computing, pp. 22–27 (2002)
6. Harville, M., Gordon, G., Woodfill, J.: Foreground segmentation using adaptive mixture model in color and depth. In: Proc. IEEE Workshop Detection and Recognition of Events in Video, pp. 3–11 (2001)
7. Eng, H., Wang, J., Kam, A., Yau, W.: Novel region-based modeling for human detection within high dynamic aquatic environment. In: Proc. IEEE Conf. CVPR, vol. 2, pp. 390–397 (2004)
8. Elgammal, A., Duraiswami, R., Harwood, D., Davis, L.S.: Background and foreground modeling using nonparametric Kernal density estimation for visual surveillance. In: Proc. of the IEEE, vol. 90(7) (July 2002)
9. Sheikh, Y., Shah, M.: Bayesian modeling of dynamic scenes for object detection. IEEE Trans. PAMI 27, 1778–1792 (2005)
10. Li, L., Huang, W., Gu, I.Y.H., Tian, Q.: Statistical modeling of complex background for foreground object detection. IEEE Trans. IP 13(11), 1459–1472 (2004)
11. Chen, T.P., et al.: Computer Vision Workload Analysis: Case Study of Video Surveillance Systems. Intel Technology Journal 09(02), 109–118 (2005)
12. Beeck, K., Gu, I.Y.H., Li, L., Viberg, M., Moor, B.D.: Region-Based Statistical Background Modelling for Foreground Object Segmentation. In: Proc. IEEE Conf. IP, pp. 3317–3320 (2006)