2010 Seventh IEEE International Conference on Advanced Video and Signal Based Surveillance

Fast Background Initialization with Recursive Hadamard Transform

Davide Baltieri, Roberto Vezzani, Rita Cucchiara
Dipartimento di Ingegneria dell'Informazione
University of Modena and Reggio Emilia
Via Vignolese, 905 - 41100 Modena - Italy
{davide.baltieri,roberto.vezzani,rita.cucchiara}@unimore.it

Abstract

In this paper, we present a new and fast technique for background estimation from cluttered image sequences. Most of the background initialization approaches developed so far collect a number of initial frames and then require a slow estimation step which introduces a delay whenever it is applied. Conversely, the proposed technique redistributes the computational load among all the frames by means of a patch by patch preprocessing, which makes the overall algorithm more suitable for real-time applications. For each patch location a prototype set is created and maintained. The background is then iteratively estimated by choosing from each set the most appropriate candidate patch, which should verify a sort of frequency coherence with its neighbors. To this aim, the Hadamard transform has been adopted which requires less computation time than the commonly used DCT. Finally, a refinement step exploits spatial continuity constraints along the patch borders to prevent erroneous patch selections. The approach has been compared with the state of the art on videos from available datasets (ViSOR and CAVIAR), showing a speed up of about 10 times and an improved accuracy.

1. Introduction

Segmentation of foreground objects using motion information is a core aspect in many computer vision systems and in particular in automated visual surveillance. Commonly, a foreground/background pixel-wise classification algorithm is adopted for each frame, relying on a background model which should be kept updated correctly. A wide variety of algorithms for background modeling and updating have been proposed [11, 13]; among the others, Mixtures of Gaussian [17] or statistical models [4] have been widely adopted. However, background initialization is still a challenging problem, in particular when all the frames contain moving objects and the empty background is never seen as a whole. Background initialization should also be very effective in cluttered video with many moving objects and fast enough to be used in real time.

Often the problem of background initialization is neglected and it is directly merged into the estimation and update steps: starting from an unreliable model, errors are identified and corrected by analyzing the extracted foreground objects. For example, in [4] a rough classification of the foreground objects is provided with a motion-based validation step and the "ghosts" (i.e., regions of apparent motion, but classified as foreground object due to a dirty background) are used to update the background. Object size, edges, optical flow or other features can be exploited to post process the detected foreground regions and to discard the erroneously detected objects [6, 10].

Broadly speaking, we can distinguish between three classes of background modeling algorithms, depending on the spatial level used. All the above described methods work at a region level, and usually are characterized by a high computational cost. Conversely, several methods try to solve the background initialization problem working at a pixel level, mainly exploiting the pixel intensity temporal constancy [8]. Even if these methods are very fast, they do not exploit spatial relations. To mitigate the problem, statistical models for each pixel [17, 18] or multiple background images have been proposed [1]. Finally, intermediate solutions work at a block (or patch) level [3, 14, 15].

Independently from the spatial level adopted, we can distinguish between recursive and non-recursive techniques. Recursive approaches maintain a single background model that is updated with each new video frame. These techniques are generally computationally efficient and have minimal memory requirements. Non-recursive techniques, instead, maintain a buffer of previous video frames and estimate a background model based solely on the statistical properties of these frames. This causes non-recursive techniques to have higher memory requirements than recursive techniques. However, since they have explicit access to the most recent video frames they can model aspects of the data

978-0-7695-4264-5/10 $26.00 © 2010 IEEE
DOI 10.1109/AVSS.2010.43

which can't be analyzed with recursive techniques.

In [3], for example, when enough frames have been collected, static blocks are selected as reference and directly included into the background; then the model is completed by iteratively adding blocks which satisfy spatial consistency and homogeneity constraints. Recently Reddy *et al.* in [14] applied a similar approach, using the DCT coefficients as a core feature to check the homogeneity constraint. These methods are general and perform very well on different types of videos. However, they are computationally too expensive to be correctly applied in real time. For example, the method by Reddy *et al.* contains an iterative block selection which prevents parallel solutions and introduces a long delay when it is executed.

In this work we propose a new fast background initialization method, working at block level in a non-recursive way, specially conceived for achieving the best background model using the minimum number of frames as possible. Similarly to [14] we split each frame into blocks, producing a history of blocks and searching among them for the most reliable ones. In this last phase, the method works at a super-block level evaluating and comparing the frequency content of each block component. Differently from [14] which makes use of the DCT coefficients, we propose the use of the Hadamard Transform [7] which is faster and particularly suitable for this type of application. In the next section we present a brief description of the Hadamard transform and it's properties. In section 3 we describe the proposed algorithm in detail, and results from real-life surveillance videos are reported in section 4.

2. The Hadamard Transform

The Hadamard transform [7, 9] belongs to the generalized class of Fourier transforms and it is based on the homonym matrix. The Hadamard matrix is a square array whose elements can be ± 1 only and its rows (and columns) are mutually orthogonal. Its recursive definition is one of the more interesting properties for our purposes. The lowest-order Hadamard matrix $\mathbf{H}_1$ has size 1x1 and it is defined as $\mathbf{H}_1 = [1]$. The Hadamard matrices having order $N = 2^n$, with n integer, can be recursively defined. In particular, given the Hadamard matrix of order N, the Hadamard matrix of order $2N$ is defined as:

$$\mathbf{H}_{2N} = \begin{bmatrix} \mathbf{H}_N & \mathbf{H}_N \\ \mathbf{H}_N & -\mathbf{H}_N \end{bmatrix} \tag{1}$$

Fig. 1 contains some example of Hadamard matrices of various orders.

A frequency interpretation of the Hadamard matrix can be given [12]. Along each row of the matrix, the frequency is related to the number of changes in sign. This frequency interpretation of the rows of a Hadamard matrix allows us

$N=2$

$$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$N=4$

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

$N=8$

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}$$

Figure 1: Hadamard matrices of order $N = 2^n$

to consider the rows to be equivalent to rectangular waves ranging between $\pm\mathbf{1}$ with a sub-period of $1/N$ units. Thus, in this context the Hadamard matrix merely performs the decomposition of a function by a set of rectangular waveforms rather than the cosine waveforms associated with the DCT.

The aforementioned structure of the Hadamard matrix shows a key feature that proves extremely useful: let $\mathbf{f}(x, y)$ be an image (or a patch extracted from it) of $2N \times 2N$ pixels. Its Hadamard transform, $\mathbf{F}(u, v)$, is given by the matrix product

$$\mathbf{F} = \mathbf{M} \cdot \mathbf{f} \cdot \mathbf{M} \tag{2}$$

where $\mathbf{M}$ is the Hadamard matrix of order $2N$.

Another way to compute $\mathbf{F}(u, v)$ is by means of the Hadamard transform of constitutive blocks. Let us decompose the image $\mathbf{f}(x, y)$ into four $N \times N$ blocks, called A, B, C, D respectively. The product of Eq. 2 can be accordingly decomposed as reported in Eq. 3:

$$\mathbf{F} = \begin{bmatrix} \mathbf{H} & \mathbf{H} \\ \mathbf{H} & -\mathbf{H} \end{bmatrix} \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{H} & \mathbf{H} \\ \mathbf{H} & -\mathbf{H} \end{bmatrix} \tag{3}$$

where $\mathbf{H}$ is the Hadamard matrix of order N. This leads to:

$$\mathbf{F} = \begin{bmatrix} \mathbf{f}_{1,1} & \mathbf{f}_{1,2} \\ \mathbf{f}_{2,1} & \mathbf{f}_{2,2} \end{bmatrix} \tag{4}$$

where

$$\begin{aligned} \mathbf{f}_{1,1} &= \mathbf{HAH} + \mathbf{HBH} + \mathbf{HCH} + \mathbf{HDH} \\ \mathbf{f}_{1,2} &= \mathbf{HAH} - \mathbf{HBH} + \mathbf{HCH} - \mathbf{HDH} \\ \mathbf{f}_{2,1} &= \mathbf{HAH} + \mathbf{HBH} - \mathbf{HCH} - \mathbf{HDH} \\ \mathbf{f}_{2,2} &= \mathbf{HAH} - \mathbf{HBH} - \mathbf{HCH} + \mathbf{HDH}. \end{aligned} \tag{5}$$

The elements **HAH**,**HBH**,**HCH** and **HDH** in Eq. 5 are the Hadamard transforms of the blocks A, B, C, D. Therefore, it's possible to compute the Hadamard transform of order **2N** from the four Hadamard transforms of order **N**, something that it's not possible in the case of the discrete cosine transform.

In addition to the recursive formulation, evaluating the Hadamard transform is faster than estimating the DCT. For example, let us consider a 8x8 block: one of the fastest DCT algorithms requires 94 multiplication and 454 additions [5], while the Hadamard transform only requires 384 integer additions for the same block (see [12] for details on the fast Hadamard computation algorithm).

Furthermore, a single 32x32 DCT (the usual size for the super-block in our algorithm) would require on average 18.560 real-arithmetic operation (multiplications and additions) [16]. By using the Hadamard transform that number is reduced to 10240 integer additions; finally by exploiting the Hadamard matrix structure by computing the 16x16 Hadamard transforms as soon as a frame is available, the number of operations required for the 32x32 transform is reduced to 3072 integer additions.

3. Recursive Hadamard Transform Background Initialization

Commonly to the other methods which estimate the background after collecting a statistically sufficient number of frames, the proposed technique analyze only the first T frames $I_t, t \in 1 \dots T$ of a video sequence. We suppose that the number T of analyzed frames is set high enough to guarantee that each part of the background is visible in one frame at least.

Similarly to [14], each frame I_t is partitioned into disjoint square blocks of 16x16 pixels. Let $b_t^{i,j}$ be the block extracted from I_t at the position (i, j). From the abovementioned assumptions, we can reasonably assert that the final background image should be obtained by the composition of opportunely selected blocks.

To this aim, a three step algorithm is proposed. In the first step, each block location is independently analyzed and a set of representative blocks is constructed for each of them. The blocks belonging to the sets which contain one element only are automatically selected and fixed as background. In other words, if a block is stable for all the first T frames, it is certainly a background block. Then, the second step aims at selecting the remaining blocks following a growing schema at a super-block level. New blocks are iteratively added to the background if they are similar enough to three neighbors which have been already included in the background.

The super-block should be constructed using the two 4-connected neighbors and the diagonal block between them.

B	C	D
A	X	E
H	G	F

Table 1: 8-connected neighbors of block X

In table 1, for example, block X can be estimated thanks to blocks A,B and C, or blocks C,D,E, and so on.

The block similarity is estimated evaluating the frequency coherence inside a super-block with the Recursive Hadamard Transform. Finally, all candidate blocks are checked again using a new refinement step to assure spatial continuity of the background image along the block borders. These three steps are described in the following subsections.

3.1. Block candidate sets

For each location (i, j), a representative set $\mathbf{R}^{i,j} = \left\{ r_k^{i,j} \right\}$ of unique blocks should be extracted. Each element $r_k^{i,j}$ is coupled with a weight $\mathbf{W}_k^{i,j}$ that denotes the number of occurrences in the image sequence. The blocks of the first frame are automatically inserted in their corresponding sets, with an initial weight set to 1. Then, for each frame $I_{2 \dots T}$ and for each location, the corresponding block is compared with all the elements of the set, looking for the most similar item. If the new block $b_t^{i,j}$ is unique (*i.e.*, it is different enough from all the other representatives) it is added to the set with weight 1; otherwise, it is used to update the most similar block in the set through a weighted mean and the corresponding weight is incremented by 1.

As proposed in [14], the similarity between two blocks b_t and r_k is checked by means of the cross-correlation (Γ) and the MAD (Φ) coefficients, respectively computed as:

$$\Gamma(r_k, b_t) = \frac{\sum_{x=1}^{N} \sum_{y=1}^{N} [r(x,y) - \mu_{r_k}] \cdot [b(x,y) - \mu_{b_t}]}{\sigma_{r_k} \sigma_{b_t}} \quad (6)$$

$$\Phi(r_k, b_t) = \frac{1}{N^2} \sum_{x=1}^{N} \sum_{y=1}^{N} \| r_k(x,y) - b_t(x,y) \| \quad (7)$$

where μ and σ are the intra-block mean and standard deviation respectively. The two blocks $\mathbf{b}_t$ and $\mathbf{r}_k$ are considered similar if

$$\Gamma(r_k, b_t) > \alpha \quad \wedge \quad \Phi(r_k, b_t) < \beta \quad (8)$$

where α and β are empirically selected; β can also be automatically estimated using the following approach (as described in [14]). Using a short training video, the MAD coefficients between co-located blocks of successive frames

are calculated. These values are sorted and only the central half of them are kept. Calling μ and σ their mean and standard deviation respectively, β can be set equal to $\mu + 2 \cdot \sigma$. This ensure that low MAD values (close or equal to zero) and high MAD values (arising due to object movements) are treated as outliers and thus ignored.

3.2. Frequency-based block selection

At time T, when all the frames have been analyzed, the background image BG can be effectively generated from the $\mathbf{R}^{i,j}$ sets. The background is firstly initialized using stable and unchanged blocks, which are characterized by $\mathbf{W}_k^{i,j} = T$. These blocks are also quickly identified from the sets with one element only. Instead, for those sets with more than one member, the background block is chosen by analyzing every representative block $r_k^{i,j}$ and comparing them with their chosen neighborhoods in the frequency domain.

The background block at the location (i, j) can be estimated if it is the only missing item in a 2x2 super-block. See fig. 2 for an example: we have to analyze the X block, given that the A, B, C blocks have been already estimated. A set of super-blocks can be constructed based on the elements of $\mathbf{R}^{i,j}$ and the known adjacent background blocks. The first super-block (called, base super-block) is constructed by forcing block X elements to zero, and filling blocks A, B, C with the known data. It's $2N$-Hadamard transform is computed from the N-Hadamard transforms of blocks A,B and C (the Hadamard transform of a null block is a null matrix), resulting in a matrix $\mathbf{C}$ of size $\mathbf{M} \times \mathbf{M}$, with $(\mathbf{M} = \mathbf{2N})$. Then, for each block in the representative set of X, a super-block is constructed by forcing A, B, C to zero and filling X with the block data. It's Hadamard transform is easily constructed, being the Hadamard transforms of blocks A, B and C a null matrices. This means that constructing the Hadamard transform of this super-block requires no arithmetic operations at all. The result is stored in a $\mathbf{M} \times \mathbf{M}$ matrix $\mathbf{D}_k$. A cost function is then defined as:

$$cost(k) = \left(\sum_{v=0}^{M-1} \sum_{u=0}^{M-1} |C(u,v) + D_k(u,v)| \right) \lambda_k \quad (9)$$

where $\lambda_k = e^{-\delta \omega_k}$, with $\delta \in [0,1]$ and $\omega_k = \mathbf{W}_k / \sum_{k=1}^{S} \mathbf{W}_k$. The representative block which yields the minimum value of the cost function is assumed to be the best candidate as background block.

3.3. Spatial continuity check and selection refinement

After all the previous steps, the chosen blocks are analyzed again to prevent some errors (as can be seen in Fig. 3a). Actually, the frequency domain constraints embedded in Eq. 9 are not enough to assure image continuity along each block border. In fact, block's candidates can have a very similar frequency content, but a different aspect, in particular this happens inside flat background parts.

Figure 2: A super-block of block X

(a) Errors in the estimated background

(b) Corrected estimated background

Figure 3: Estimated background before and after block selection correction

The average gradient along each border is computed, and if the average gradient of two or more sides is greater than a given threshold γ, the selected block is discarded and a new background block is selected among the remaining unique representative blocks of the corresponding set, using the same algorithm described in section 3.2. The threshold value γ has been empirically set using part of the CAVIAR dataset [2] (which yields the most errors in our experiments). Given the sets of correctly and erroneously

selected blocks, γ is the mean value of the corresponding intra-set average gradients.

This final refinement allows to further improve the estimation accuracy.

4. Experimental results

We carried out our experiments on a total of 144 surveillance videos as reported in Table 2. The starting frame of each sequence was chosen accordingly to avoid trivial conditions like uncluttered scenes and background-only frames. Few videos were also resized to 384×288 from their original dimensions to obtain comparable results.

RHT was compared with the DCT-based algorithm presented in [14] and a trivial median filter approach. The three methods have been implemented in C++, partially using the OpenCV libraries and the Imagelab processing libraries, and we carried on tests on a 1.6 GHz dual core processor.

Parameters were set as follows: block size was set to $\mathbf{16x16}$ pixels, α was empirically set to $\mathbf{0.8}$ as well as δ set to $\mathbf{0.5}$, β was automatically set to $\mathbf{10}$ for the CAVIAR dataset, and $\mathbf{4}$ for the ViSOR dataset, γ was set to $\mathbf{60}$.

The DCT algorithm was able to elaborate images at 33 fps, but the actual background computation was done at the end of the process in 1506 ms (on average). This time is not compliant with a real time processing and the delay introduced after the frame T is considerable. The median background has the same drawback, with an even greater delay due to the sorting algorithm which should be performed on all the frames. Using the proposed approach, instead, the computational load is more balanced, images are processed at 27 fps, while the actual background computation at the end takes on average 97ms only. Table 3 shows timing results for the three algorithms.

To evaluate objectively the resulted background images we used a methodology similar to the one presented in [6], but extended to color images. In order to compare the results from the three algorithms the average error (AE) and the number of clustered error pixels (CEP) are used. AE is the average of the distances between the pixels of the estimated and true background, while CEP is the number of error pixels that are 4-connected to other error pixels. A pixel of the estimated background is considered an error pixel if the distance from the same pixel of the true background is greater than 20. Table 4 shows the averaged results for the CAVIAR dataset [2], while table 5 shows results for the ViSOR dataset [19].

Fig. 4 shows example results from four video sequences, the first two from the ViSOR dataset and the last two from the CAVIAR dataset.

Table 3: Timing results

	Frame Update (ms)	Background Estimation (ms)
Median	46	3133
DCT-based Method [14]	29	1506
RHT	**36**	**97**

Table 4: Averaged results using CAVIAR dataset

	Average error	Clustered error pixels
Median	16.00	5451
DCT-based Method [14]	14.12	3822
RHT	**12.55**	**2334**

Table 5: Averaged results using ViSOR dataset

	Average error	Clustered error pixels
Median	11.080	1929
DCT-based Method [14]	13.55	1807
RHT	**12.62**	**968**

5. Conclusions

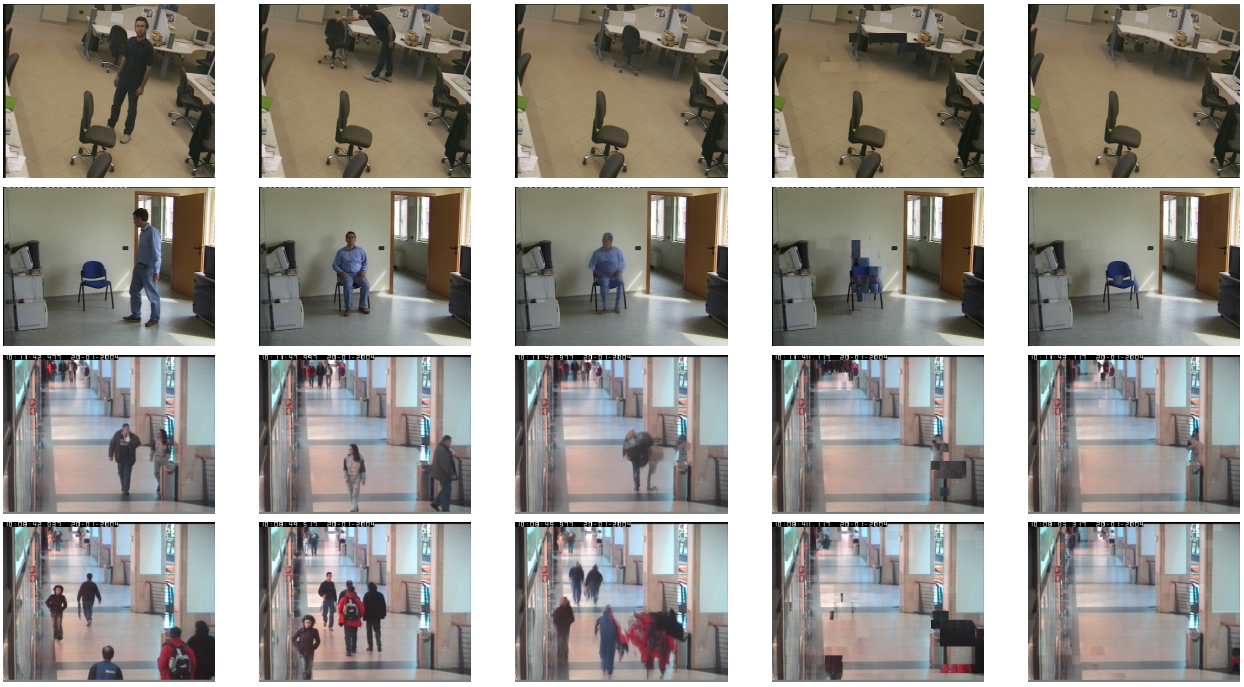
We proposed a new background estimation algorithm which can be effectively used for the initialization step. Frames are processed in a block wise manner and for each block the best candidate among the ones coming from all the analyzed frames is selected. To this aim, we firstly fix non variable blocks as background; than, the holes left by moving objects are filled in selecting those blocks whose frequency spectrum is coherent with the already fixed neighbors.

Differently from similar approaches which make use of DCT coefficients to check the block similarity, we adopt the Hadamard transform, which is as effective as the Discrete Cosine Transform, but its computational load is lower and, above all, can be split in frame by frame pre-processing tasks. In such a manner, the final step of block selection does not introduce any particular delay and the frames can be processed in real time. Finally, the spatial coherence along the block borders is also checked and leads to block rejection if it is not verified.

The method has been tested on different surveillance videos, coming from some of the most popular dataset (e.g,. ViSOR [19] and CAVIAR [2]). Moreover, it was added in our surveillance framework both for background initialization at the start phase and for background reset whenever required thanks to its real time performances.

Table 2: Datasets Summary

Dataset	Subset	Num. of Videos	Description	Frames (Start - End)	Size	Parameters
CAVIAR	set 1	28	large indoor room	100 (300-400)	384*288	$\alpha = 0.8, \delta = 0.5$ $\beta = 10, \gamma = 60$
CAVIAR	set 2cor	26	indoor corridor side view	100 (0-100)	384*288	$\alpha = 0.8, \delta = 0.5$ $\beta = 10, \gamma = 60$
CAVIAR	set 2front	26	indoor corridor front view	100 (0-100)	384*288	$\alpha = 0.8, \delta = 0.5$ $\beta = 10, \gamma = 60$
ViSOR	Outdoor Unimore D.I.I. setup - Multicamera	28	outdoor large space	100 (100-200)	384*288	$\alpha = 0.8, \delta = 0.5$ $\beta = 4, \gamma = 60$
ViSOR	Indoor Domotic Unimore D.I.I. setup	16	indoor small room	100 (100-200)	384*288	$\alpha = 0.8, \delta = 0.5$ $\beta = 4, \gamma = 60$
ViSOR	Video for indoor people tracking with occlusions	6	indoor small room	100 (100-200)	384*288	$\alpha = 0.8, \delta = 0.5$ $\beta = 4, \gamma = 60$
ViSOR	Outdoor Unimore D.I.I. setup - Multicamera - disjoint views	14	outdoor large space	100 (200-300)	384*288	$\alpha = 0.8, \delta = 0.5$ $\beta = 4, \gamma = 60$



A B C D E

Figure 4: Examples from two VISOR and two CAVIAR videos: (A,B) two random frames, (C) Estimated background using the median filter, (D) using the DCT based method of Reddy *et al.* ([14]), (E) Our proposed enhanced method

Acknowledgments

This work has been done within the THIS project with the support of the Prevention, Preparedness and Consequence Management of Terrorism and other Security-related Risks Programme European Commission - Directorate-General Justice, Freedom and Security.

References

[1] A. Bevilacqua, L. Di Stefano, and A. Lanza. An effective multi-stage background generation algorithm. In *Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance,*, pages 388–393, Los Alamitos, CA, USA, 2005. IEEE Computer Society.

[2] Ec funded caviar project/ist 2001 37540. Website. http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/.

[3] A. Colombari, A. Fusiello, and V. Murino. Background initialization in cluttered sequences. In *CVPRW '06: Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop*, page 197, Washington, DC, USA, 2006. IEEE Computer Society.

[4] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati. Detecting moving objects, ghosts and shadows in video streams. *IEEE Trans. on PAMI*, 25(10):1337–1342, Oct. 2003.

[5] E. Feig and S. Winograd. Fast algorithms for the discrete cosine transform. *IEEE Transactions on Signal Processing*, 40(9):2174 –2193, sep 1992.

[6] D. Gutchess, M. Trajkovic, E. Cohen-Solal, D. Lyons, and A. Jain. A background model initialization algorithm for video surveillance. In *Proc. of IEEE Intl Conference on Computer Vision*, volume 1, pages 733–740, 2001.

[7] J. Hadamard. Resolution d'une question relative aux determinants. *Bull. Sci. Math.*, 17:240–246, 1893.

[8] W. Long and Y.-H. Yang. Stationary background generation: an alternative to the difference of two images. *Pattern Recogn.*, 23(12):1351–1359, 1990.

[9] R. Paley. On orthogonal matrices. *J. Math. Phys.*, 12:311–320, 1933.

[10] D. H. Parks and S. S. Fels. Evaluation of background subtraction algorithms with post-processing. In *AVSS '08: Proceedings of the 2008 IEEE Fifth International Conference on Advanced Video and Signal Based Surveillance*, pages 192–199, Washington, DC, USA, 2008. IEEE Computer Society.

[11] M. Piccardi. Background subtraction techniques: a review. In *Proceedings of the IEEE International Conference on Systems, Man & Cybernetics*, pages 3099–3104, oct 2004.

[12] W. Pratt, J. Kane, and H. Andrews. Hadamard transform image coding. *Proc. of the IEEE*, 57(1):58–68, Jan. 1969.

[13] R. Radke, S. Andra, O. Al-Kofahi, and B. Roysam. Image change detection algorithms: a systematic survey. *IEEE Transactions on Image Processing*, 14(3):294 –307, march 2005.

[14] V. Reddy, C. Sanderson, and B. Lovell. An efficient and robust sequential algorithm for background estimation in video surveillance. In *Proc. of IEEE Int'l Conference on Image Processing*, pages 1109–1112, 2009.

[15] D. Russell and S. Gong. A highly efficient block-based dynamic background model. In *Proceedings of IEEE Conference on Advanced Video and Signal Based Surveillance. AVSS 2005.*, pages 417 – 422, sept. 2005.

[16] H. Sorensen, D. Jones, M. Heideman, and C. Burrus. Real-valued fast fourier transform algorithms. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 35(6):849 – 863, jun 1987.

[17] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. In *Proc. of IEEE Int'l Conference on Computer Vision and Pattern Recognition*, pages 246–252, 1999.

[18] T. Tanaka, A. Shimada, D. Arita, and R. ichiro Taniguchi. A fast algorithm for adaptive background model construction using parzen density estimation. In *AVSS '07: Proceedings of the 2007 IEEE Conference on Advanced Video and Signal Based Surveillance*, pages 528–533, Washington, DC, USA, 2007. IEEE Computer Society.

[19] R. Vezzani and R. Cucchiara. Video surveillance online repository (visor): an integrated framework. *Multimedia Tools and Applications*, Oct. 2009.