Multimed Tools Appl (2015) 74:11099–11115
DOI 10.1007/s11042-014-2219-4

Anomaly detection in compressed H.264/AVC video

Sovan Biswas · R. Venkatesh Babu

Received: 16 August 2013 / Revised: 18 July 2014 / Accepted: 24 July 2014/
Published online: 28 August 2014

Abstract Real time anomaly detection is the need of the hour for any security applications. In this article, we have proposed a real time anomaly detection for H.264 compressed video streams utilizing pre-encoded motion vectors (MVs). The proposed work is principally motivated by the observation that MVs have distinct characteristics during anomaly than usual. Our observation shows that H.264 MV magnitude and orientation contain relevant information which can be used to model the usual behavior (UB) effectively. This is subsequently extended to detect abnormality/anomaly based on the probability of occurrence of a behavior. The performance of the proposed algorithm was evaluated and bench-marked on UMN and Ped anomaly detection video datasets, with a detection rate of 70 frames per sec resulting in 90× and 250× speedup, along with on-par detection accuracy compared to the state-of-the-art algorithms.

Keywords Anomaly detection · H.264 · Motion vectors · Compressed domain video analysis · Kernel density estimation · Visual surveillance

1 Introduction

Analysis of crowd behavior has been a point of focus for various vision research groups for decades due to the increasing importance to physical security. The requirement of crowd behavior analysis spans from anomaly detection in surveillance scenarios to global crowd pattern analysis for safety planning in highly populated regions, of which former is of particular interest to security personnel. As ‘A stitch in time saves nine’ is the main motive for security personnel, the focus is on real-time anomaly detection in surveillance videos. With more and more locations being covered with surveillance cameras, the problem of real-time analysis and monitoring becomes a huge challenge.

S. Biswas · R. V. Babu (✉)
Video Analytics Lab, Supercomputer Education and Research Center, Indian Institute of Science, Bangalore 560012, India
e-mail: venky@serc.iisc.ernet.in

Computer vision with recent advances is able to solve the potential problem of anomaly detection to large extent with sufficient accuracy. New algorithms are being developed to improve the accuracy of anomaly detection [2, 4, 8, 11]. Adam et al. [1] use histograms to characterize optical flow in a patch. Kratz and Nishino [10], harness spatio-temporal gradient to detect the abnormality. Kim et al. [9] models local optical flow with Mixtures of Probabilistic Principal Component Analyzers (MPPCA) and enforce the local consistency using Markov Random Field. On the other hand, Mehran et al. [12] propose Social force concept. Mahadevan et al. [11] tries mixture of dynamic textures (MDT) [5] to model normal crowd behavior. Wu et al. [26] utilize particle trajectories to model crowd behavior.

But, most of the algorithms are computationally expensive and thus not suitable for real time applications. As majority of the current algorithms work with fully decompressed videos having pixel level information, there is an apparent need of decompressing the video before processing. Apart from delay due to video decompression from any compressed format (MPEG-2, MPEG-4, etc.), huge amount of data and more complex feature extraction in pixel domain leads to slower execution speed. The problem of decompression overhead is trivial for few short videos or online processing of a feed from a single camera, but becomes humongous for large scale anomaly detection, both online and offline on a single centralized system. For example, a building being surveilled 24×7 by a single CCTV camera, recording at 25 frames per second, results in 24×3600×25 = 2160000 frames per day. During offline processing, assuming decompressing algorithms achieve 100 to 300 frames per second to decode, one would require 7200 to 21600 seconds $\approx$ 2 to 6 hours only to decode 24 hours of video! Even in online processing, decoding overhead becomes a concern when hundreds of video feeds are processed simultaneously for anomaly detection on a centralized system. The focus of this paper is to reduce these computational overheads by performing compressed video anomaly detection. Even though the accuracy is compromised to some extent, this can provide fast initial screening for pixel domain analysis on uncompressed videos for better accuracy.

H.264 [25] is the recent and the best video compression standard due to its high compression factor for the given video quality. It is currently one of the widely used video compression standard especially in visual surveillance application. As with any compression standard, majority of compression is achieved by removing the temporal redundancy between neighboring frames using Motion Vectors (MVs). MVs contain the shift of macroblocks across two consecutive frames thus saving bits required to encode the similar blocks across frames. In crude sense, MVs are considered to be a coarse approximation of optical flow. But in H.264, MVs are more accurate than any previous standards because of use of variable block-size motion compensation and quarter-pel motion estimation technique. Variable block-size motion compensation supports motion prediction for 16×16 to 4×4 block sizes, enabling precise segmentation and motion prediction. The supported prediction block sizes include 16×16, 16×8, 8×16, 8×8, 8×4, 4×8, and 4×4 that can be used together in a single macroblock. Half pel and quarter pel motion prediction are subsequently used to predict block motion with better accuracy, resulting in nearly optical flow like characteristics.

MVs, along with compression, also help in various video analysis tasks such as anomaly detection [3]. Macroblocks corresponding to anomalous regions behave differently with respect to neighboring macroblocks, both temporally and spatially. Specifically, MVs corresponding to anomalous macroblocks exhibit different characteristics. The proposed algorithm computes the statistics of magnitude and orientation of MVs in macroblock through probability of their occurrence at a particular region to detect anomaly.

The rest of the article is divided in 4 sections. Section 2 begins with a brief discussion on other proposed algorithms for anomaly detection. Subsequently, Section 3 presents the proposed algorithm in detail followed by experimentation and results in Section 4. The paper is concluded with a short conclusion in Section 5.

2 Related work

Video anomaly detection approaches can be roughly classified into two categories, namely trajectory analysis and motion analysis. In trajectory analysis, normal behavior is learned through trajectory patterns and interaction of tracked objects [7, 15, 19, 22]. As these sets of algorithms heavily depend on tracking, it is not suitable for crowded scenes. On the other side, motion analysis involves abnormality detection based on the local patterns of the motion. The proposed work can be categorized in the later category, where we estimate the abnormality based on the motion.

In one of the pioneer works involving optical flow, Adam et al. [1] used multiple local monitors which characterized optical flow through histograms. Anomaly was detected through careful integration of multiple alerts from different monitors. Kim et al. [9] captured the distribution of the typical optical flow through Mixture of Probabilistic Principal Component Analyzers (MPPCA), then used Markov Random Field (MRF) to enforce local consistency. Mehran et al. [12] proposed Social force concept compared to Lagrangian trajectories, by Wu et al. [26], on optical flow, to define the crowd movement. Ryan et al. [20] proposed textures of optical flow and measured the uniformity of the flow field to detect abnormality. As all the above methods involved optical flow computation, they failed to cater the requirement of real time solutions despite of they being well suited for variety of the anomaly problems providing accurate solutions.

Many methods in recent times were developed bypassing the optical flow computation and relied on object size and appearance. Mahadevan et al. [11] proposed mixture of dynamic textures (MDT) [5], which jointly modeled the appearance and dynamics of crowd scenes. Though it was a comparatively effective methodology, processing each frame and extraction of textures resulted in huge computation and low execution speed. To overcome the same, recently Reddy et al. [18] proposed dividing the frame into non overlapping blocks for individual processing. This was followed by detection based on features generated through combination of motion, size and texture of the region. Cong et al. [6] used sparse reconstruction cost (SRC) to evaluate the normalness of the testing sample by using the normal bases (which is local spatio-temporal patches obtained from training normal videos).

All the aforementioned set of algorithms mainly involve a video sequence containing normal behavior for training and testing sequence having abnormal activity. Recently, unsupervised approach for anomaly detection in crowds was proposed by Sun et al. [23], which used attractive motion disorder concept, a linear combination of motion saliency and block motion.

The proposed method is different from others in the respect that we model the normal behavior for each location individually via pyramidal approach to reduce the computation. The following section presents the proposed algorithm in detail.

3 Algorithm

Anomaly is defined by the departure from usual characteristics. Mathematically, let $y = [x_1, \ldots x_n]$ be a candidate event at a particular region/location, where x_i is its i^{th} feature. But in actuality, an event is characterized by its features. So, the probability of occurrence of an event directly depends on the probability of occurrence of its features, i.e., $P(y) = P(x_1, \ldots, x_n)$. Without loss of generality, we can assume independence among its features, resulting in

$$P(y) = \prod_{i=1}^{n} P(x_i)$$

where $P(y)$ is the probability of occurrence of the event. Thus, anomaly is defined as

$$P(y) \leq \tau \tag{1}$$

where τ is the decision threshold.

In general, a behavior could be normal in one feature space but abnormal in another. This reduces the problem to extract relevant features that could discriminate the usual from unusual behaviors.

Typically in any crowd video, crowd motion is consistent during normal scenario. They tend to have a distinctive motion patterns/features. Representing the motion through its magnitude and orientation, together, is found to be adequate to learn the usual pattern and detect anomaly. For example, a person riding a cycle on footpath has different magnitude than that of a person walking, even though direction of motion is same. On the other side, in case of sudden suspicious activity, there is sudden change in crowd movement magnitude and orientation than usual.

Motion Vectors (MVs) in H.264/AVC are defined for a minimum of 4 × 4 block, which reduces the analysis computation by sixteen times than pixel level processing. But with increase in video resolutions, computation increases proportionally. Though handling high resolution videos are computationally expensive, it can be processed effectively by utilizing pyramidal representation. So, one can begin processing at coarser level (reduced frame size) and in case of ambiguity, move to finer level (actual frame size) to resolve it. This hierarchical processing leads to different levels of MVs representation (created using smoothing and sub-sampling), which we refer as Motion Pyramid (Fig. 1).

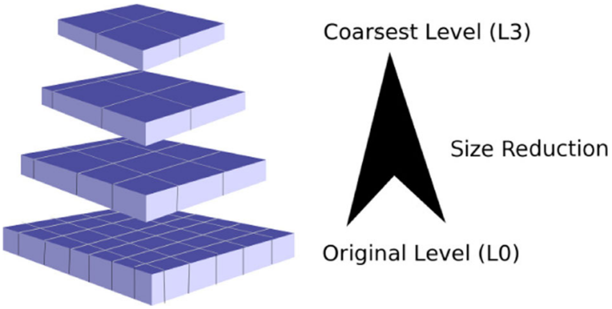


Fig. 1 Motion Pyramid with 4 Levels from L0 to L3

In this article, we tried to tap the local variations in magnitude and orientation of MVs (feature) to classify normal and abnormal events by modeling usual behavior for each location. The major modules of the proposed algorithm includez: a) Modeling usual behavior b)Detecting abnormal pattern.

3.1 Modeling usual behavior

The proposed algorithm is trained for usual observations either from training videos or from the initial frames of each videos containing usual behavior patterns. Training is performed for each pyramidal level to learn the usual behavior pattern. Primarily, training is divided into two stages : i) Pre-processing and ii) Usual behavior modeling.

3.1.1 Pre-processing

Motion Vector (MV) Replication H.264/AVC encoding is performed on variable block size for better prediction and reduction in overall bit rate. So, MVs are estimated for variable blocks depending upon the content of the video. Instead of handling the variable block size MVs explicitly, we replicated the MVs for higher size macroblocks to its 4 × 4 constituents resulting in MVs for every 4 × 4 blocks. This without any loss of content creates matrix of uniform size which are easier to handle (Refer Fig. 2b). Additionally assuming linearity in the motion prediction, if current P frame is encoded using any past k previously encoded frames as reference, existing literature [13, 16] suggested rescaling MVs proportionally with respect to the distance between the reference frame and current P frame in the video sequence.

Finer Level ($L0$) As MVs are aimed at reducing the amount of bits required for encoding a specific macroblock, it does not always reflect the true object motion. In order to minimize the effect of noise in MVs, a spatio-temporal median filter is applied on the MVs as shown in Fig. 2c. Additionally, median filtering guarantees motion interpolation for I-frames. For effective filtering, temporal range is divided into equal amount of past and future frames.

$$x\left[m, n, t\right] = median\left\{\tilde{x}\left[p, q, r\right], (p, q, r) \epsilon w\right\} \tag{2}$$

where, x and $\tilde{x}$ are the filtered MVs and raw MVs respectively. w represents a neighborhood centered around location (m, n, t) in the spatio-temporal cube. As median filtering is a computationally expensive step, we have used a small cube of 5 × 5 × 5.

Coarser Level (Reduced Size) (L_i) Coarse Level (L_i) is obtained from previous level by spatial bi-linear interpolation of higher resolution raw MVs of each complete frame to one

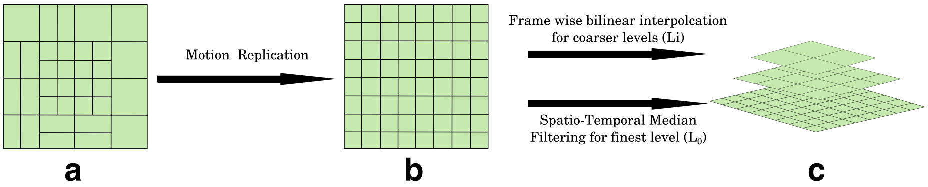

Motion Replication
Frame wise bilinear interpolcation for coarser levels (Li)
Spatio-Temporal Median Filtering for finest level (L$_0$)
a
b
c

Fig. 2 **a)** Original variable macroblocks present in a single frame **b)** Motion replication upto its 4 × 4 constituent macroblocks. **c)** Motion pyramid with reduction in frame size fourth of previous level size (keeping in tandem with minimum macroblock size of 4). Though MVs are noisy, interpolation reduces the noise drastically making additional filtering futile. However, to have temporal consistency in coarser levels, MVs are averaged out temporally at each location. Furthermore, averaging ensures approximate motion estimation for I-frames at coarser levels (See Fig. 2c).

3.1.2 Usual behavior (UB) modeling

As behavior is characterized by MVs, UB modeling is done by forming histograms of motion magnitude and orientation at each pyramidal level L. Typically, MVs vary from one region to another in a frame. For example, MVs for regions away from the camera exhibit lower motion magnitude than those near to camera. The variation is not only limited to motion magnitude but also the motion orientations, as direction of movement at each location can be very different and depends upon topography spanned by camera. Marginalized histograms are formed from temporal observation of both motion magnitude and orientations, respectively. We deliberately refrained from joint statistics as joint histograms at every location at a particular pyramidal level L would require large amount of memory for storage.

Though ideally, capturing the motion variations should result in true statistics. But due to insufficient and inconsistent information at each location in training videos, histograms are noisy and need to be corrected. For example in Fig. 3a, locations P_0, P_1 and P_2 have different statistics. P_0 has high density of information compared to others whereas P_2 has the least density. Recent research in approximate nearest neighbor fields (ANNF mapping [17]) based on coherency have demonstrated that two neighboring location tend to exhibit same property with high probability. Therefore, accumulated statistics are further refined by interpolating based on neighbor characteristics. A Balloon estimator (Adaptive Kernel Density Estimator (AKDE)) is applied to achieve interpolation to form individual histograms. Since,$P\left(x_{i,j,k\pm a}\right)$ and $P\left(x_{i\pm b,j\pm b,k}\right)$ have effect on $P\left(x_{i,j,k}\right)$, we use Gaussian kernel in AKDE (3) both spatially and across histogram bins with a variable kernel size. Here $x_{i,j,k}$ denotes count of motion magnitude and orientation at spatial location $[i, j]$ falling in k^{th} bin.

$$f_h(z) = \frac{1}{n}\sum_{l=1}^{n} K_h(z - z_l) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{z - z_l}{h}\right) \tag{3}$$

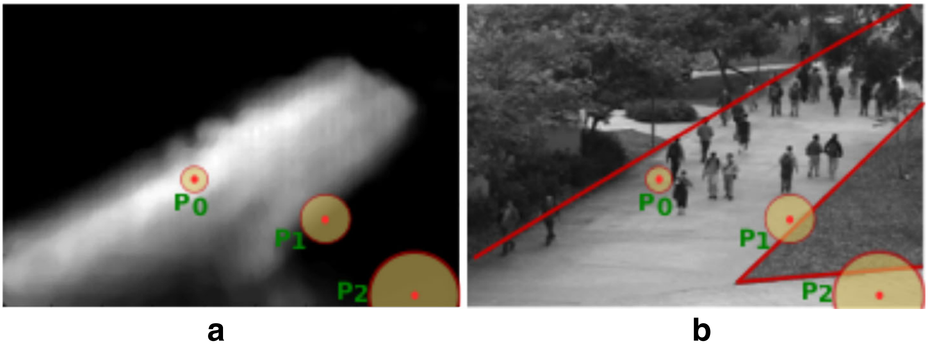


Fig. 3 **a)** Sample density for the scene **b)** Variable kernel size at three different points P_0, P_1 and P_2. P_0 has highest number of training samples followed by P_1 and P_2. Thus, the kernel size of P_2 is the biggest followed by P_1 and P_0

where,
$$K(\bullet) = (2\pi)^{-k/2} | \Sigma |^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{z}-\mu)^T \Sigma^{-1}(\mathbf{z}-\mu)\right),$$
$$k = 3, \mu = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \Sigma = \begin{pmatrix} \sigma_s^2 & 0 & 0 \\ 0 & \sigma_s^2 & 0 \\ 0 & 0 & \sigma_b^2 \end{pmatrix}, h = \frac{c}{g(z)}$$

$K(\bullet)$ is 3D Gaussian kernel with σ_s spatial standard deviation and σ_b across histogram range. z being the data. h is the variable kernel size based on the density of the samples $g(z)$ at that location. c (set to 1) and n are scaling factor for bandwidth h and number of samples respectively. This is based on the assumption that any highly used regions by crowd will have large number of samples to capture proper statistics. So, it is intuitive to use a smaller kernel around the high density region compared to other regions. (Refer Fig. 3b).

Marginalized histograms, for motion magnitude and orientation, are subsequently normalized to obtain respective probability density function. UB probability is computed using (1), where motion magnitude and orientation are two features.

$$P(y) = P(x_{\text{magnitude}}) * P(x_{\text{orientation}}) \tag{4}$$

So at each pyramidal level L and location p, the anomaly is defined as

$$P(y_L(p)) \leq \tau_L(p)$$

where, $y_L(p)$ and $\tau_L(p)$ denote the event and descision threshold at location p and pyramidal level L respectively.

Due to varying topography captured by the field of view of camera, the motion statistics (motion magnitude and orientation) change from one location to another. Thus, at each pyramidal level (L), the decision threshold τ_L varies based on locations. So instead of computing different thresholds for each location p through trial and error, we further define $\tau_L(p)$ based on a constant value.

Let, $\mathcal{H}(p) = \{h_u(p)\}_{u=1\ldots m}$ be the l_1 normalized histograms of a usual behavior feature at location p and u being the bin values. Then,

$$\mathcal{V}(p) = sort(\{h_u(p)\}_{u=1\ldots m}) \quad \%\ descending\ order$$

where, $\mathcal{V}(p)$ denotes the sorted probability (histogram) values at location p. Subsequently,

$$\begin{aligned} b^* &= \arg\min_b \left(\left\{ \sum_{i=1}^{b} \mathcal{V}_i(p) \right\} \geq \gamma_L \right) \quad \text{where, } b = 1\ldots m \\ \tau_L(p) &= \mathcal{V}_{b^*}(p) \end{aligned}$$

where, $\sum_{i=1}^{b} \mathcal{V}_i(p)$ denotes the cumulative sum; b is the bin number. γ_L is a constant value for all locations at a particular pyramidal level L. (Refer Fig. 4 for details). In case of multiple features,

$$\tau_L(p) = \prod_{j=1}^{n} \mathcal{V}_{b^*}^{j}(p) \tag{5}$$

where, n is the number of features. Here, $n = 2$ as features are motion magnitude and orientation. Thus, the probability of an event being anomaly is now defined as

$$P\left(y_L(p)\right) \leq \tau_L(p) = \mathcal{F}(\mathcal{H}(p), \gamma_L) \tag{6}$$

where, $\mathcal{F}$ is the above explained function.

In real time scenarios, the training set is accompanied by validation set. Unlike the training set that only has positive instances, the validation set consists of both positive and

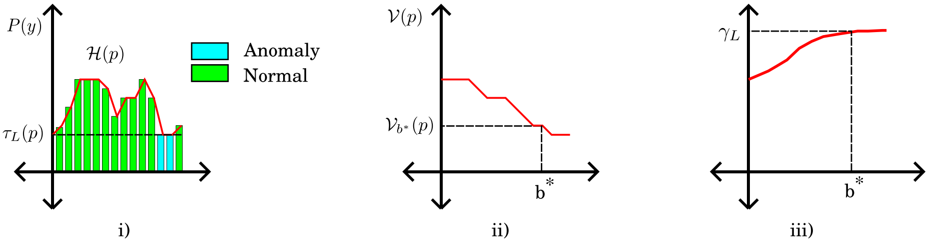


Fig. 4 **i)** Usual behavior histogram **ii)** Sorted histogram values **iii)** Plot of cumulative sum of the sorted histogram values used to find b^* based on γ_L

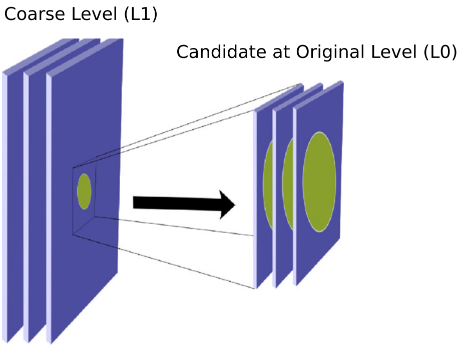
negative instances for estimating the decision parameters γ_L for all levels. Usually, the majority of selection or rejection of candidates for anomaly detection occur at coarsest scale (Lf, where f is the coarsest level) followed by finer levels in motion pyramid. So, we propose a greedy strategy to estimate the value for γ_{Lf} followed by other intermediate levels according to motion pyramid. Using extreme levels of motion pyramid (ie. level Lf and $L0$), the value of γ_{Lf} is fixed by maximizing anomaly detection on the validation set. Later each level i from level $f-1$ to level 1 is added to the motion pyramid and corresponding γ_{Li} is fixed by maximizing anomaly detection on the validation set with previously fixed γ values for coarser level.

3.2 Detecting abnormal pattern

The use of motion pyramid plays a major role in reducing the computation without affecting the quality. In a nutshell, detection is started at coarsest level and moves to finer level only if anomaly is suspected at current level (see Fig. 5). Thus effectively, it is a single frame detection without any relation to future and past frames.

In the proposed approach, features (motion magnitude and orientation) of each location of a test video frames are compared to usual behavior to detect the region of anomaly. But in a bid to increase detection rate in test videos, raw MVs are pre-processed only at coarsest level (similar to training phase) and detection is performed based on probability of occurrence of an event at every location, frame-by-frame. Any, pre-processing (filtering) and detection at finer pyramid levels are delayed till a probable anomalous candidate region is suspected at previous coarser level (Fig. 5). To further reduce computation, preprocessing (filtering) is performed only at suspected candidate locations. Anomaly is suspected at

Fig. 5 Detection from level L1 to L0. Anomaly (*circle*) is suspected at coarse level L1 is further processed at finer level L0



location p if $P(y_{Li}(p)) \leq \tau_{Li}(p)$, where, $y_{Li}(p)$ represents probability of occurrence of an event at coarse level (Li) and $\tau_{Li}(p)$ acts as decision threshold computed using γ_{Li}. In case an anomalous candidate is found at a particular location in previous level, preprocessing and detection is performed only at that location at current level. This hierarchical processing across levels results in huge computational savings. At finest level $(L0)$ and location p, $P(y_{L0}(p)) \leq \tau_{L0}(p)$ indicates abnormality (Ref. (1)), where, $y_{L0}(p)$ and $\tau_{L0}(p)$ represents event and decision thresholds respectively. The proposed steps could still result in some amount of spectral noise. These are anomalous detections mainly at the edges of moving body due to improper orientation. As our goal is to find a consistent anomaly movement rather than a false detection in a single frame, we additionally perform median filtering on $P(y_{L0}(p))$. This ensures that the anomaly is detected consistently rather than detecting a noisy motion boundary.

The proposed pyramidal approach provides a great boost up for high resolution videos as majority of the computation is bypassed at coarser levels. In low resolution videos, processing is done only at original level as coarser levels fail to capture the motion magnitude variations satisfactorily at coarser level, leading to many mis-detection.

4 Experiments and results

In this section, we introduce the datasets used for evaluation as well as to describe the evaluation procedure. We have conducted experiments on three video databases to demonstrate the capability of the proposed algorithm to handle wide range of variations with comparable accuracy to the state-of-the-art techniques. Since these datasets were not encoded in H.264 format, we encoded the same in H.264 format using baseline profile (only I and P frames) with 1 reference frame and Group of Pictures (GOP) length is set to 30. Baseline profile is ideal for network cameras and video encoders since low latency is achieved because of the absence of B-frames (http://www.axis.com/products/video/about_networkvideo/compression_formats.htm). We have used x264 and $JM15.1$ for encoding and partial decoding of the compression parameters respectively. All the experiments were performed using MATLAB on single core 3.4 GHz processor.

4.1 Evaluation procedure

Algorithm is evaluated on two aspects namely global anomaly detection and localized anomaly detection. Ped1 [14] (HD videos) is used to test global and localized anomaly detection, whereas Ped2 [14] and UMN [24] crowd datasets are used for global anomaly detection only.

Ped1 contains training set of 34 clips, whereas Ped2 has 16 sets of clips. The testing set consist of 36 clips for Ped1 and 12 clips for Ped2. We have increased the size of Ped1 and Ped2 datasets from 158 × 238 and 240 × 320 to 480 × 720 respectively. This is done to validate our pyramidal approach. Anomalies are divided into two categories a) Non-pedestrians among the pedestrians and b) Pedestrians moving into unusual regions. The aim is to detect abnormality in a frame and localize the corresponding regions.

Similar to Mahadevan et al. [11], evaluation is performed on Ped1 for two aspects: frame level anomaly detection and pixel level anomaly detection (anomaly localization), whereas only frame level anomaly detection for Ped2. a) *Frame Level Anomaly*: A frame is anomaly (true positive) if at least one pixel of the frame is detected anomalous in comparison to a certain threshold. The threshold is subsequently varied to determine ROC curve. b) *Pixel Level Anomaly*: On the other side, anomaly localization is measured by comparing the detected anomaly of the given algorithm with respect to the ground truth anomalous region for each frame. If at least 40 % of the truly abnormal locations are detected, then the frame is considered to be anomalous (True positive). Whereas, a frame is considered false positive if ground truth indicates it to be normal but one or more of its pixels are detected as anomalous. The decision threshold is varied to obtain anomaly localization ROC curve, as mentioned above.

UMN dataset consist of single video with 11 sequences of 3 different scenes of frame size 320 × 240, where an abrupt crowd anomaly occurs at the end of each sequence. The first scene has 2 sample sequences, whereas second and third scenes have 6 and 3 sample sequences respectively. The basic intention is to detect abnormal frames. The ground truth abnormal frames are marked through tags at the upper left corner.

4.2 Quantitative performance analysis

The proposed algorithm is capable of achieving real-time prediction by compromising some accuracy. The number of pyramidal level generated differs from videos to videos but regulated by the restriction on size of the coarsest level to a minimum of 30 × 30. Other parameters are set for different datasets are explained below.

Ped1 and Ped2 Typically, the statistics (motion magnitude and orientation histograms) are different for each location p at a particular pyramidal level L, so varying $\tau_{Li}(p)$ is defined using a constant γ_{Li} (6). Currently, the proposed algorithm uses two levels of pyramids for Ped1 and Ped2, where coarse level is obtained by reducing the finer level by a fraction of 4. Thus, there are two decision thresholds γ_{L1} and γ_{L0}. The anomaly evaluation is performed by varying γ_{L0} to obtain receiver operating characteristic (ROC). This evaluates the effect of γ_{L0} on the anomaly detection. But to evaluate effect γ_{L1} on the output, we studied the variation of the γ_{L1} on Equal Error Rate (EER) rate of Ped1 (Fig. 6 and Table 1). Thus, we set γ_{L1} to 0.99 for Ped1 and Ped2.

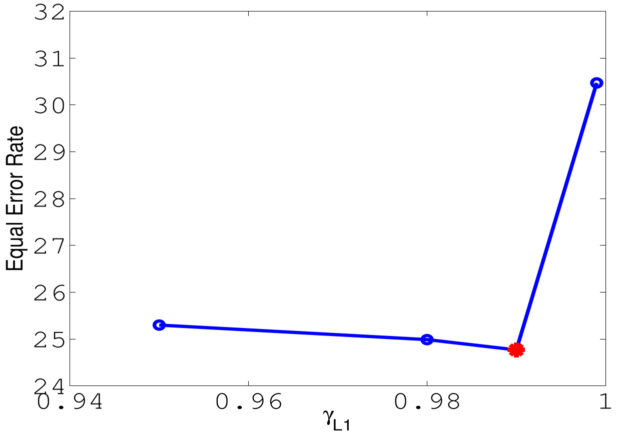
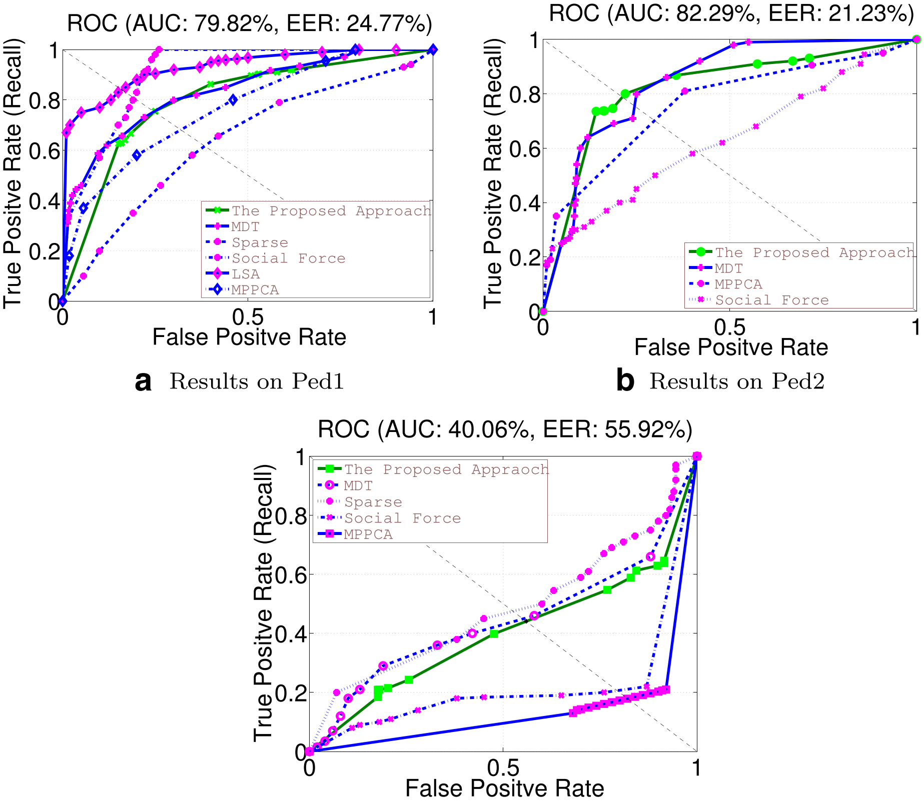


Fig. 6 Equal Error Rate (*EER*) vs γ_{L1} plot for Ped1 dataset. '*' denotes optimal minimum error while varying the thresholds

Table 1 Variation of Equal Error Rate with varying γ_{L1}

γ_{L1}	EER	Detection Rate
0.999	30.47	90 fps
0.99	**24.77**	**70 fps**
0.98	24.99	65 fps
0.95	25.30	58 fps

The proposed algorithm has achieved frame-level anomaly detection of equal error rate (EER) 24.77 % on Ped1 (Fig. 7a) and 21.23 % on Ped2 (Fig. 7b), which is comparable to Cong et al. [6] and MDT [11] (Table 2), with better localization of anomaly than any existing methods. This is demonstrated by rate of detection of 44.08 % on Ped1 (Fig. 7c). Refer Table 3 for comparisons. Additionally, it is able to reduce computational complexity resulting in real time detection. We have achieved around 70 frames per sec for 480 × 720 resolution video.



a Results on Ped1

b Results on Ped2

c Anomaly Localization on Ped1 with Pixel GroundTruth

Fig. 7 ROC curve for the two datasets

Table 2 Equal Error Rate (EER) on Ped datasets

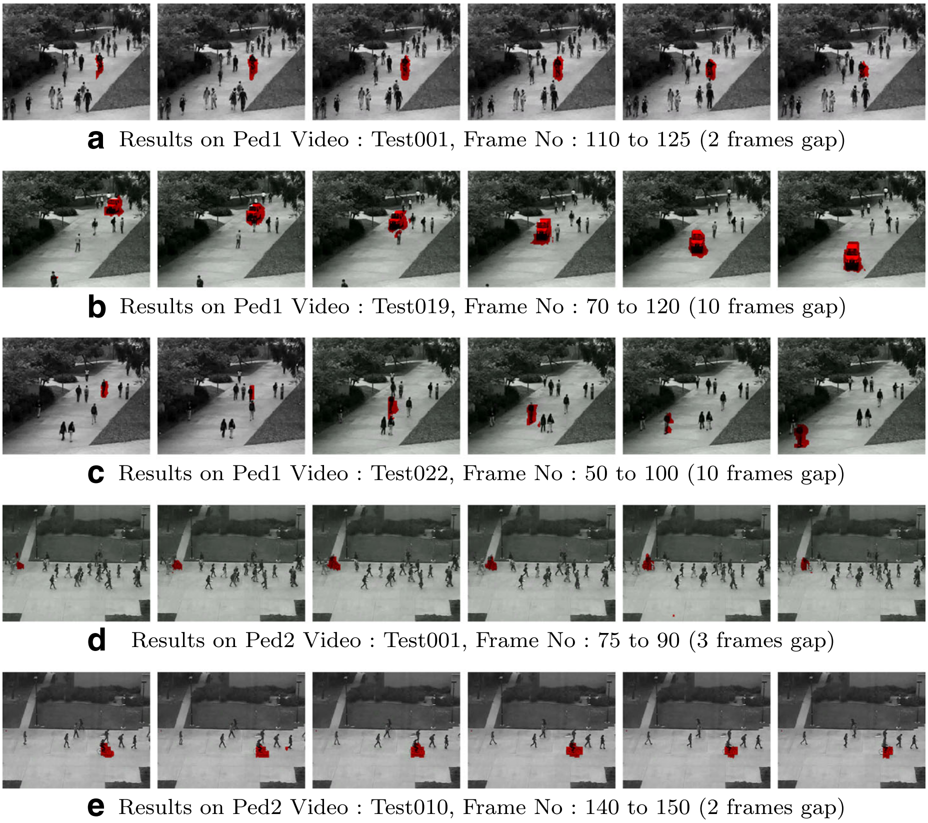
Approaches	Ped 1	Ped 2
SF [11, 12]	31 %	42 %
MPPCA [9, 11]	40 %	30 %
SF-MPPCA	32 %	36 %
MDT [11]	25 %	25 %
Sparse [6]	19 %	-
LSA [21]	16 %	-
Ours	**24.77** %	**21.23 %**

In comparison to 0.04 frames per sec by MDT [11], and comparison to 0.25 frames per sec by sparse approach [6], we have achieved around 1700× speedup and 250× speedup respectively. Few of the sample results are shown in Fig. 8 .

UMN dataset Before divulging into experiments on UMN dataset, we observed some interesting points about UMN sequences and its abnormal frame marking. In Fig. 9a, though abnormality has already begun but the ground truth indicates otherwise. Additionally, in Fig. 9b, the scene is empty (static frame), whereas ground truth indicates abnormality. On evaluating the algorithm to detect global anomaly detection on UMN dataset, we observed that the proposed algorithm is able to detect all the abnormality but by a shift, which was expected because of rationale discussed earlier. Since the feature extraction of the proposed algorithm depends on the MVs, it fails to detect anomaly if there is no motion. We thus modified the ground truth by considering static frames as normal and accommodating other aforementioned reasons. There are 11 video sequences in UMN dataset. For training we have used first sequence of each scene and left the remaining sequences in the same scene for testing. For learning usual behavior model, we have only relied upon normal frames and neglected the abnormal frames. The evaluation is done on each scene independently. Interestingly, each of these sequences have movements only corresponding to particular region while training but tend to move in other regions during testing. On comparing our results with original ground truth we have achieved ROC curve with area under the curve (AUC) 68.67 %, but with corrected ground truth we achieved AUC of 94.28 %.

Table 3 Rate of Detection (RD), Area Under the curve (AUC) and Detection speed for Ped 1

Approaches	RD	AUC	Detection speed
SF [11, 12]	21 %	17.9 %	-
MPPCA [9, 11]	18 %	20.5 %	-
SF-MPPCA	18	21.3 %	-
MDT [11]	45 %	44 %	0.04 fps
Sparse [6]	46 %	46.1 %	0.25 fps
Ours	**44.08 %**	**40.06 %**	**70 fps**


a Results on Ped1 Video : Test001, Frame No : 110 to 125 (2 frames gap)

b Results on Ped1 Video : Test019, Frame No : 70 to 120 (10 frames gap)

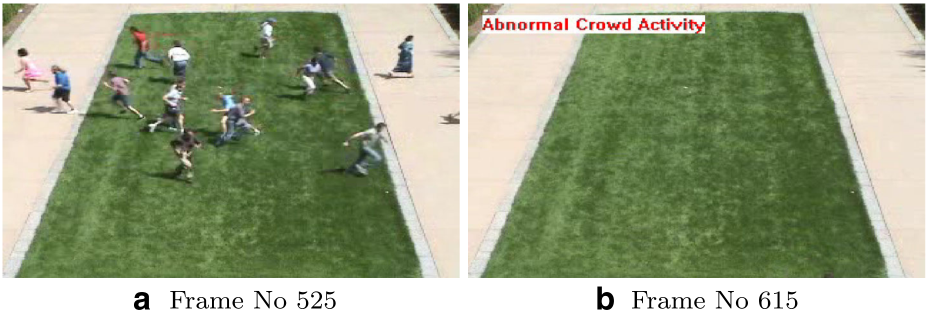
c Results on Ped1 Video : Test022, Frame No : 50 to 100 (10 frames gap)

d Results on Ped2 Video : Test001, Frame No : 75 to 90 (3 frames gap)

e Results on Ped2 Video : Test010, Frame No : 140 to 150 (2 frames gap)

Fig. 8 Results on Ped1 and Ped2

(Refer Fig. 10 and Table 4). Since, resolution of UMN dataset is low, we have processed only at finer level ($L0$). Even then, computationally we achieved around 70 frames per sec (a speedup of 90× compared to Sparse approach [6]). Fig. 11 shows detection on UMN-



a Frame No 525

b Frame No 615

Fig. 9 Frames of UMN dataset wrongly marked **a)** Abnormal frame marked as normal frame **b)** normal frame marked as abnormal frame

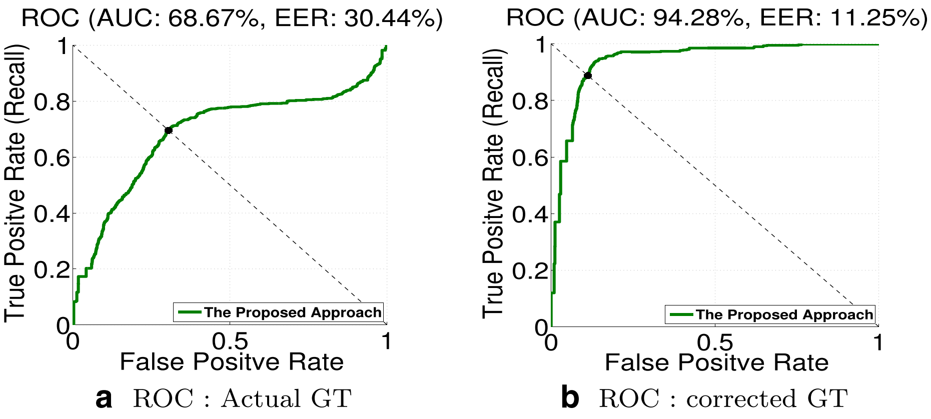


Fig. 10 ROC curve for UMN datasets **a)** Based on original ground truth **b)** Based on modified ground truth

Table 4 Comparative results with other approaches on UMN

Approaches	ROC
Social Force [12]	96
Nearest Neighbor [6]	93
Sparse [6]	97.8
Ours Without GT correction	68.67
Ours - With GT correction	**94.28**

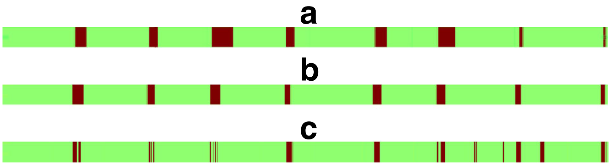
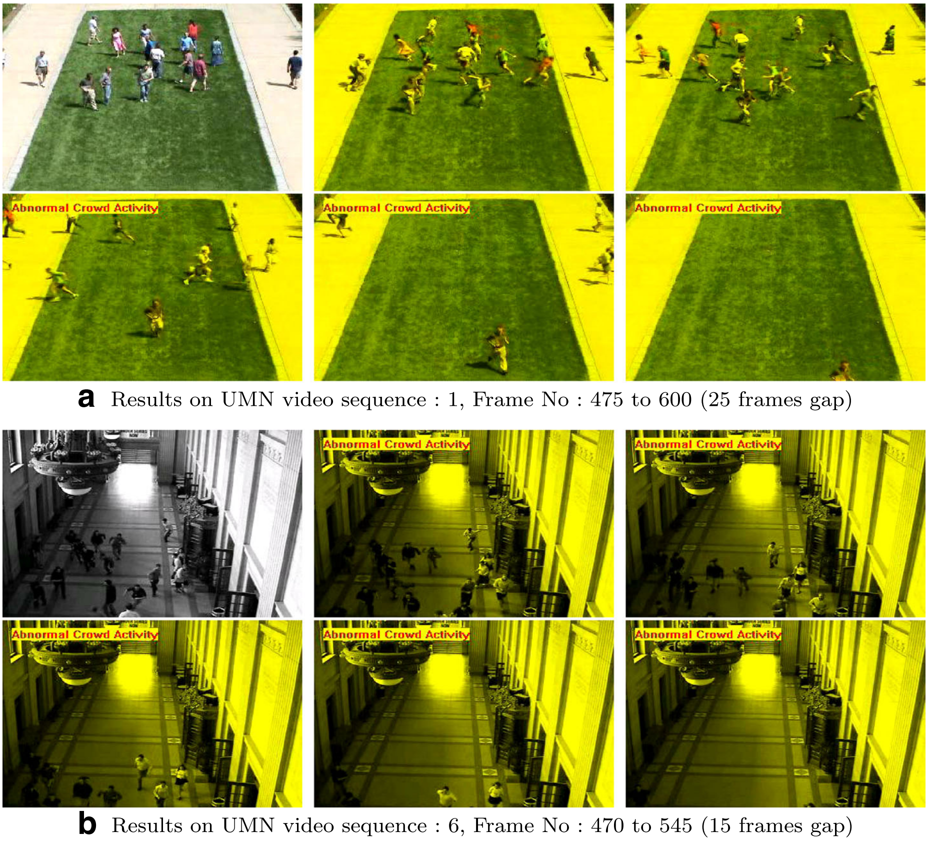


Fig. 11 Result comparison on UMN dataset : Labels of each test frame **a)** Original GT bar **b)** Modified GT bar **c)** Actual detection bar; Green : Normal frame, Red : Abnormal frame



a Results on UMN video sequence : 1, Frame No : 475 to 600 (25 frames gap)

b Results on UMN video sequence : 6, Frame No : 470 to 545 (15 frames gap)

Fig. 12 Results on UMN dataset (*Yellow colored Frames are abnormal Frames*)

dataset. The occurrence of few false positives are due to unavailability of training samples at unused regions during training. Few of detection results on UMN dataset are shown in Fig. 12.

5 Conclusion and future work

We have proposed a compressed domain approach in H.264/AVC framework to detect anomalies in surveillance videos. MV magnitude and orientation together contain enough information for anomaly detection, which was tapped by profiling MVs through Motion Pyramid. The proposed method performs on-par compared to state-of-the-art results with huge reduction in computation time resulting in more than real-time performance. Even though initial results are encouraging, the effect of other compression parameters can be studied further along with exploring the opportunity to harness temporal movements (such as loitering) for further improvement.

References

1. Adam A, Rivlin E, Shimshoni I, Reinitz D (2008) Robust real-time unusual event detection using multiple fixed-location monitors. IEEE Trans Pattern Anal and Mach Intell 30(3):555–560
2. Benezeth Y, Jodoin PM, Saligrama V, Rosenberger C (2009) Abnormal events detection based on spatio-temporal co-occurences. In: Proceedings of the IEEE conference on Computer vision and pattern recognition, pp 2458–2465
3. Biswas S, Babu RV (2013) Real time anomaly detection in H.264 compressed videos. In: National conference on Computer vision, pattern recognition, image processing and graphics. NCVPRIPG, pp 1–4
4. Boiman O, Irani M (2005) Detecting irregularities in images and in video. In: Proceedings of the IEEE international conference on Computer vision, pp 462–469
5. Chan AB, Vasconcelos N (2008) Modeling, clustering, and segmenting video with mixtures of dynamic textures. IEEE Trans Pattern Anal and Mach Intell 30(5):909–926
6. Cong Y, Yuan J, Liu J (2011) Sparse reconstruction cost for abnormal event detection. In: Proceedings of the IEEE conference on Computer vision and pattern recognition, pp 3449–3456
7. Hu W, Xiao X, Fu Z, Xie D, Tan T, Maybank S (2006) A system for learning statistical motion patterns. IEEE Trans Pattern Anal and Mach Intell 28(9):1450–1464
8. Itti L, Baldi P (2005) A principled approach to detecting surprising events in video. In: Proceedings of the IEEE conference on Computer vision and pattern recognition, vol 1, pp 631–637
9. Kim J, Grauman K (2009) Observe locally, infer globally: a space-time MRF for detecting abnormal activities with incremental updates. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2921–2928
10. Kratz L, Nishino K (2009) Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models. In: Proceedings of the IEEE conference on Computer vision and pattern recognition, pp 1446–1453
11. Mahadevan V, Li W, Bhalodia V, Vasconcelos N (2010) Anomaly detection in crowded scenes. In: Proceedings of the IEEE conference on Computer vision and pattern recognition, pp 1975–1981
12. Mehran R, Oyama A, Shah M (2009) Abnormal crowd behavior detection using social force model. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 935–942
13. Moiron S, Faria S, Assunçao P, Silva V, Navarro A (2007) H. 264/AVC to MPEG-2 video transcoding architecture. In: Proceeding of the conference on Telecommunications, pp 449–452
14. Peds dataset. http://www.svcl.ucsd.edu/projects/anomaly/dataset.html
15. Piciarelli C, Micheloni C, Foresti GL (2008) Trajectory-based anomalous event detection. IEEE Trans Circ and Syst for Video Technol 18(11):1544–1554
16. Pourazad MT, Nasiopoulos P, Ward RK (2010) Generating the depth map from the motion information of H.264-encoded 2D video sequence. J Image and Video Process 2010(4):4:1–4:13
17. Ramakanth SA, Babu R. V (2012) Feature match: an efficient low dimensional patchmatch technique. In: Proceedings of the indian conference on computer vision, graphics and image processing, pp 45:1–45:7
18. Reddy V, Sanderson C, Lovell BC (2011) Improved anomaly detection in crowded scenes via cell-based analysis of foreground speed, size and texture. In: IEEE conference on Computer vision and pattern recognition workshops. CVPRW, pp 55–61
19. Remagnino P, Jones GA (2001) Classifying surveillance events from attributes and behaviour. Algorithms 6(7):1–12
20. Ryan D, Denman S, Fookes C, Sridharan S (2011) Textures of optical flow for real-time anomaly detection in crowds. In: 8th IEEE international conference on Advanced video and signal-based surveillance. AVSS, pp 230–235
21. Saligrama V, Chen Z (2012) Video anomaly detection based on local statistical aggregates. In: Proceedings of the IEEE conference on Computer vision and pattern recognition, pp 2112–2119
22. Stauffer C, Grimson WEL (2000) Learning patterns of activity using real-time tracking. IEEE Trans Pattern Anal and Mach Intell 22(8):747–757
23. Sun X, Yao H, Ji R, Liu X, Xu P (2011) Unsupervised fast anomaly detection in crowds. In: Proceedings of the ACM international conference on Multimedia, pp 1469–1472
24. UMN dataset. http://mha.cs.umn.edu/proj_events.shtml#crowd
25. Wiegand T, Sullivan GJ, Bjontegaard G, Luthra A (2003) Overview of the H.264/AVC video coding standard. IEEE Trans Circ and Syst for Video Technol 13(7):560–576
26. Wu S, Oreifej O, Shah M (2011) Action recognition in videos acquired by a moving camera using motion decomposition of lagrangian particle trajectories. In: Proceedings of the IEEE international conference on Computer vision, pp 1419–1426

Sovan Biswas is currently pursuing M.Sc. (Engg.) degree at Video Analytics Laboratory, Supercomputer Education and Research Centre, Indian Institute of Science, Bangalore, India. He received his Bachelor of Engineering (Computer Science) in 2009 from Visveswariah Technological University, Belgaum, India. Later, he worked in IBM India Software Labs, Bangalore. He joined Supercomputer Education and Research Centre, Indian Institute of Science, for MSc (Engg.) in 2011. His research interests include compressed video analysis, crowd analysis and machine learning.

R. Venkatesh Babu received his Ph.D. degree in electrical engineering from the Indian Institute of Science, Bangalore, India, in 2003. He held post-doctoral positions with the Norwegian University of Science and Technology, Norway, and with IRISA/INRIA, Rennes, France, through ERCIM fellowship. Subsequently, he was a Research Fellow with Nanyang Technological University, Singapore. He spent couple of years working in the industry. He is currently an Assistant Professor and convenor of Video Analytics Laboratory at Supercomputer Education and Research Centre, Indian Institute of Science, Bangalore, India. His research interests include video analytics, human-computer interaction, computer vision, and compressed domain video processing. He is a senior member of IEEE.