Efficient Video Encoding for Automatic Video Analysis in Distributed Wireless Surveillance Systems

LINGCHAO KONG and RUI DAI, University of Cincinnati

In many distributed wireless surveillance applications, compressed videos are used for performing automatic video analysis tasks. The accuracy of object detection, which is essential for various video analysis tasks, can be reduced due to video quality degradation caused by lossy compression. This article introduces a video encoding framework with the objective of boosting the accuracy of object detection for wireless surveillance applications. The proposed video encoding framework is based on systematic investigation of the effects of lossy compression on object detection. It has been found that current standardized video encoding schemes cause temporal domain fluctuation for encoded blocks in stable background areas and spatial texture degradation for encoded blocks in dynamic foreground areas of a raw video, both of which degrade the accuracy of object detection. Two measures, the sum-of-absolute frame difference (SFD) and the degradation of texture in 2D transform domain (TXD), are introduced to depict the temporal domain fluctuation and the spatial texture degradation in an encoded video, respectively. The proposed encoding framework is designed to suppress unnecessary temporal fluctuation in stable background areas and preserve spatial texture in dynamic foreground areas based on the two measures, and it introduces new mode decision strategies for both intra- and interframes to improve the accuracy of object detection while maintaining an acceptable rate distortion performance. Experimental results show that, compared with traditional encoding schemes, the proposed scheme improves the performance of object detection and results in lower bit rates and significantly reduced complexity with comparable quality in terms of PSNR and SSIM.

CCS Concepts: • **Information systems** → **Multimedia information systems**; • **Computer systems organization** → *Sensor networks*; • **Hardware** → *Sensor applications and deployments*; • **Computing methodologies** → Computer vision;

Additional Key Words and Phrases: Video encoding, video analysis, surveillance systems, wireless systems

ACM Reference format:
Lingchao Kong and Rui Dai. 2018. Efficient Video Encoding for Automatic Video Analysis in Distributed Wireless Surveillance Systems. *ACM Trans. Multimedia Comput. Commun. Appl.* 14, 3, Article 72 (July 2018), 24 pages.
https://doi.org/10.1145/3226036

1 INTRODUCTION

Wireless embedded camera sensors are playing crucial roles in various distributed surveillance applications such as border patrol, traffic monitoring, and environmental monitoring. In many

This work was supported by the National Institute of Standards and Technology under Grant 60NANB17D193 and the National Science Foundation under Grant CNS-1644946.
Authors' addresses: L. Kong, Department of Electrical Engineering and Computer Science, University of Cincinnati, Cincinnati, OH 45221; email: konglo@mail.uc.edu. R. Dai, Department of Electrical Engineering and Computer Science, University of Cincinnati, Cincinnati, OH 45221; email: rui.dai@uc.edu.
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2018 ACM 1551-6857/2018/07-ART72 $15.00
https://doi.org/10.1145/3226036

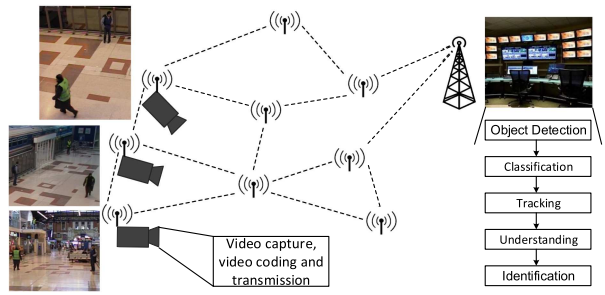


Fig. 1. Typical architecture of distributed wireless surveillance systems.

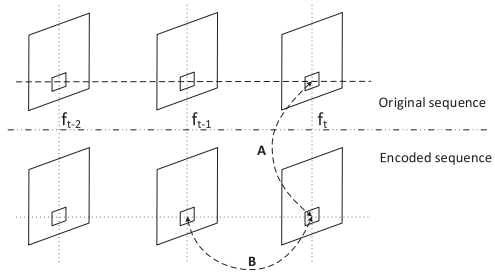


Fig. 2. Schematic diagram of encoding distortion calculation.

distributed wireless surveillance systems [35], camera sensors report their video observations to a central base station through wireless communication; the typical architecture is shown in Figure 1. Due to the low computing power and limited energy and bandwidth on embedded cameras, raw videos acquired by camera sensors are usually preprocessed, encoded, and compressed before being delivered to the base station [40]. A powerful central server or a data center at the base station can fully utilize its powerful computing capability and perform data fusion from multiple cameras to obtain a much better understanding of the surveillance videos than individual cameras [13, 31].

A typical automatic surveillance system includes the following stages: object detection, classification of objects, tracking, understanding and description of behaviors, and final human identification [13]. Object detection is the first and the most essential step of the entire procedure, because detecting objects provides a focus of attention for later processes such as tracking and behavior analysis. However, the inevitable degradation of video quality caused by lossy compression at embedded cameras has a significant impact on object detection [16, 20]. Therefore, video encoders for surveillance systems should be designed to improve the performance of object detection.

The block-based hybrid approach (intra-/interpicture prediction and 2D transform coding) is employed in all modern video compression standards such as H.264/AVC [38] and the latest HEVC. As shown in Figure 2, this approach measures the encoding distortion by comparing the encoded video with the original video (*A* direction) using the metric *SSD*, namely, *Sum of Squared Differences*, which is obtained by the sum of squared differences of the intensity between the encoded video and the original video in the macroblock (MB) unit. This strategy can result in two problems: (1) temporal domain fluctuation in the encoded video (*B* direction in Figure 2) when colocated regions of consecutive frames (e.g., f_{t-1} to f_t) are not consistently encoded, especially when intraframes are periodically inserted at low and medium bitrates, and (2) spatial texture degradation, since *SSD* could not effectively reflect the degradation status of spatial texture. In our preliminary

work [19], we have studied the effects of lossy compression on object detection in depth, and we have found that the temporal domain fluctuation in stable background areas and the spatial texture degradation in dynamic foreground areas degrade the accuracy of object detection in a compressed video.

In this article, we propose an efficient video encoding framework for distributed wireless surveillance systems with the objective to improve the performance of object detection on compressed videos. The proposed framework uses the sum-of-absolute frame difference (SFD) to depict the temporal domain fluctuation, and the degradation of texture (TXD) to quantify the degree of spatial texture degradation in an encoded video. Both measures have been demonstrated to be highly correlated with the accuracy of object detection in our previous work [19]. For the encoding of background areas in a raw video, we introduce a *Temporal-Fluctuation-Reduced video Encoding* scheme (TFRE) based on the SFD, and for the encoding of dynamic foreground areas, we introduce a *Spatial-Texture-Preserved video Encoding* scheme (STPE) based on the TXD in the 2D transform domain (TXD^{SIT}). Both schemes are standard compliant, in which new mode decision strategies are incorporated in the standardized encoding procedure to optimize the performance of object detection. Our preliminary results on the TFRE scheme have been presented in our recent work [17, 18]. Unique contributions of this article include (1) the STPE scheme is designed based on a new spatial textual descriptor in the 2D transform domain, which is presented for the first time in this article; (2) the STPE scheme is integrated with the TFRE scheme to a standard-compliant video encoding framework; (3) in addition to the original dataset in our preliminary work, a new dataset is introduced to evaluate the proposed algorithms; and (4) using both the original dataset and the new dataset, the performance of the proposed algorithms is thoroughly evaluated in terms of computational complexity, pixel-level detection accuracy, and object-level detection accuracy.

The rest of this article is organized as follows. In Section 2, we review the related video encoding algorithms in the literature. In Section 3, we investigate systematically the impact of lossy compression on object detection. Based on these findings, we propose the efficient standard-compliant video encoding framework in Section 4. In Section 5, the performance of the proposed framework is evaluated. Finally, we present concluding remarks in Section 6.

2 RELATED WORK

There exist several encoding algorithms especially designed for improving the performance of object detection. In [1], regions of individual frames containing high-frequency spatial features, corners, and edges, which are detected by FAST and Sobel detectors, are preserved while other regions are smoothed in the encoding process. For efficient video processing and analysis in the compressed domain, a coding method is proposed that optimizes the accuracy of motion information embedded in a code stream based on the affine motion model [28]. In [22], two typical usages of task-based video, license plate recognition and medical diagnosis, are studied, and a task-based video quality optimization approach is proposed, which is driven by object recognition rates during the encoding process. A model of human detection accuracy based on the object area and video compression ratio is established in [5], and based on this model, an appropriate amount of bitrate is allocated to each moving camera in mobile surveillance networks. Although these existing encoding algorithms could improve the performance of object detection, they have not addressed the problem of temporal fluctuation in background areas, which can reduce the accuracy of object detection.

On the other hand, the problem of temporal fluctuation has been investigated with the objective to improve the perceptual quality of compressed videos. The temporal fluctuation perceived by humans is defined as *flicker*, which usually refers to frequent luminance or chrominance perceptual changes that do not appear in uncompressed raw videos [15]. A temporal low-pass filtering scheme

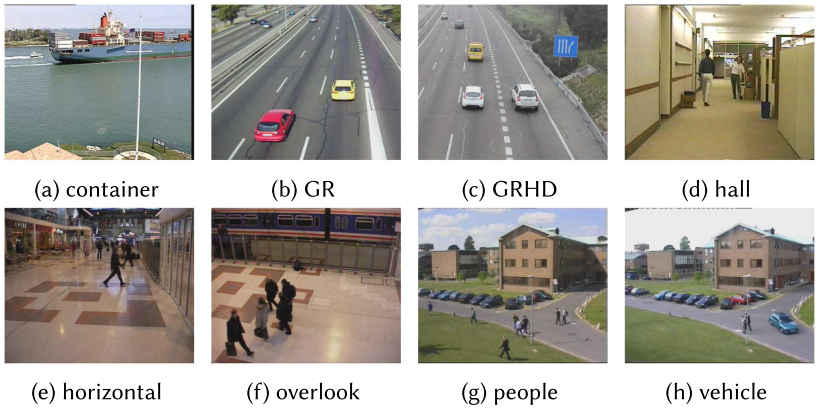


Fig. 3. Snapshots of video sequences.

is proposed that smooths the luminance changes on a block-by-block basis in [15]. A two-pass coding scheme is proposed in [39], which involves a first pass of simplified P-frame coding to derive a no-flicker reference of the current frame, and a second pass of actual I-frame coding with small QPs for closely approaching the no-flicker reference. A modified distortion measure that considers the distortions in both *A* and *B* directions in Figure 2 to reduce flicker is applied during the intraprediction mode rate distortion optimized selection process in [6]. For the flicker artifact in HEVC, a region-classification-based rate control for Coding Tree Units in I-frames is proposed to improve the reconstructed quality of I-frames to suppress flicker in [36]. Different from these methods that are designed to optimize human visual perception, our proposed work addresses the temporal fluctuation problem to improve the performance of object detection. It is worthwhile to address this problem since the human vision system and the computer vision system may have different responses to an encoded video.

The conservation of spatial texture has been studied in several video encoding solutions. A region-based rate control scheme for better subjective quality is proposed in [12], in which each frame is first divided into complex textural regions, flat regions, and moving regions, based on their interframe rate distortion behaviors. Then, the regions containing complex textures are treated as one basic unit for rate control. In [41], a perspective motion model is employed to warp static textures and utilize texture synthesis to encode dynamic textures, which results in bitrate savings at the same video quality. For facilitating visual retrieval, textural features in spatial domains, such as gradient-based features like SIFT and SURF, are better preserved by designing specific rate control strategies in [4]. An HEVC framework of jointly compressing the visual feature descriptors and video content is proposed for visual retrieval in [42], in which the high-efficiency coding is achieved by exploiting the interactions between video features and visual content. While the purposes of these methods are to improve either objective and subjective quality or the performance of visual retrieval, our proposed work utilizes the relationship between spatial texture and the 2D transform encoding to preserve spatial texture for the better performance of automatic video analysis.

3 THE IMPACT OF LOSSY COMPRESSION ON OBJECT DETECTION

We constructed a distorted video database to study the impact of lossy compression on the performance of object detection [19]. Eight video sequences with different spatial and temporal details were chosen. The snapshots of these videos are shown in Figure 3. Among them, *container*, *GR*, and

GRHD are typical test videos for traffic monitoring; *hall*, *horizontal,* and *overlook* are indoor scenes; and *people* and *vehicle* are outdoor scenes. The open-source H.264/AVC encoder x264 [25] was used to compress the raw videos. The one-pass constant QP mode was applied in the x264 encoder, and the length of GOP was set to 20 with the IPPP structure. Each raw video was compressed using 19 different QPs ranging from 22 to 40, which resulted in a total number of 152 compressed videos.

Object detection algorithms can be classified into two main groups: optical flow and background subtraction [13, 37]. Background-subtraction-based object detection algorithms attract the most attention due to their high accuracy and moderate complexity. As suggested in [32], background subtraction algorithms can be summarized into several categories based on their principles. Three algorithms from different categories were selected to be executed on the compressed videos: the Gaussian Mixture Model (GMM) algorithm from the statistical category, the algorithm that combines statistical background estimation and per-pixel Bayesian segmentation (referred to as the GMG algorithm) from the nonparametric category, and the Adaptive Background Learning (ABL) algorithm from the basic category.

One is unlikely to have prior knowledge on what object detection algorithm will be used by a certain surveillance application. Therefore, it is hard to estimate the absolute accuracy of object detection at the encoder side. We propose to estimate the relative performance of object detection on compressed videos in comparison to uncompressed raw videos. Object detection results from the raw videos are regarded as *Ground Truth* (GT), and results from the compressed videos are *Algorithm Results* (AR). *Recall* and *Precision* are common metrics to evaluate the performance of object detection [2]. *Recall* denotes the percentage of correctly detected foreground pixels in the total foreground pixels in GT, and *Precision* denotes the ratio of correctly detected foreground pixels to the total number of pixels detected in AR, which are given by

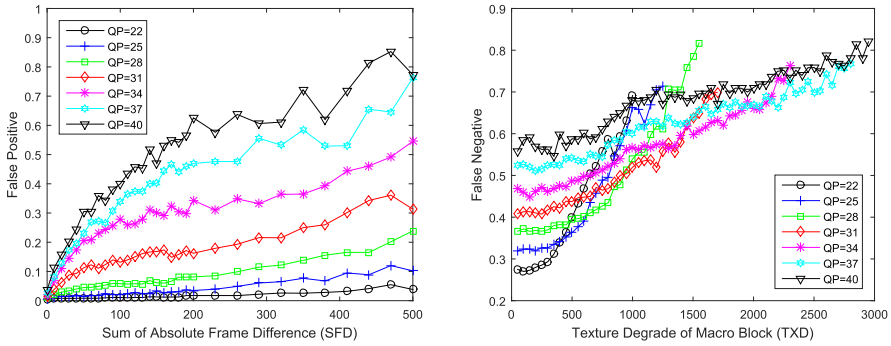
$$\begin{aligned} Recall &= \frac{TP}{TP + FN} \\ Precision &= \frac{TP}{TP + FP}, \end{aligned} \tag{1}$$

where *TP*, *FN*, and *FP* stand for the amount of true-positive pixels, false-negative pixels, and false-positive pixels, respectively. Since *Recall* and *Precision* selectively assess the level of missing TP and mistaking TP, it is hard to evaluate the performance of algorithms using one of these metrics alone. Therefore, the overall performance of detection algorithms could be measured by their harmonic mean F_1 [2], which is given by

$$F_1 = 2 \cdot \frac{Recall \cdot Precision}{Recall + Precision}. \tag{2}$$

Based on the definition of Recall and Precision, the accuracy of object detection is related to TP, FN, and FP, and the union of TP and FN is the ground truth, a constant value if given the raw video and the detection algorithm. Therefore, it is sufficient to characterize the relative object detection performance of a compressed video by estimating the average values of FP and FN over different detection algorithms. To enable such estimation, in the following analysis, we consider the scenario that a coarse-grained classification of foreground and background MBs can be done on the encoder side. A simple frame differencing method was applied before encoding to label coarse-grained foreground and background MBs in our distorted video database.

Unlike human beings who can easily extract and focus on a moving object from a blurred background, the performance of computer vision algorithms can be affected by the quality of the background. The background should be stable in the temporal domain to facilitate object detection; however, the procedure of video coding might introduce temporal fluctuations of the background



(a) (b)

Fig. 4. The trends of FP versus SFD and FN versus TXD.

that can cause FP. We conducted statistical analysis on the relationship between temporal fluctuations and FP [19]. To describe the degree of temporal fluctuation in stable background areas, we have introduced *SFD*, *the Sum-of-absolute Frame Difference in MB unit between the current frame and the previous frame*, which is given by

$$SFD = \sum_{i,j=1}^{i,j=16} |m_t(i,j) - m_{t-1}(i,j)|, \tag{3}$$

where $m_t(i,j)$ is the reconstructed pixel value at location (i,j) in an MB of the current frame and $m_{t-1}(i,j)$ is the reconstructed pixel value in the corresponding MB of the previous frame.

To understand how SFD is related to FP, we applied three different object detection algorithms on a large number of videos that have different content characteristics and were compressed under different quantization steps [19]. We collected SFD and FP samples from stable background areas in all the encoded videos. The relationships between FP and SFD for the three detection algorithms are similar, and the averages for the three algorithms are shown in Figure 4(a). From the figure, we can find that FP grows when SFD increases, and higher compression (larger QP) can result in higher FP levels, indicating that FP is closely associated with SFD.

Edge and texture are the key elements for object detection. If there is no clear boundary between the foreground and the background, it is difficult to detect an object accurately. After comparing *algorithm results* with *ground truth* in our entire dataset, we find that foreground areas with large texture deterioration are highly likely to be detected as FN. Based on this phenomenon, we introduce *TXD* [19], *the absolute difference of texture in MB unit between the encoded frame and the original frame*, to describe texture degradation:

$$TXD = \left| \sum_{i,j=1}^{i,j=16} g_t(i,j) - \sum_{i,j=1}^{i,j=16} G_t(i,j) \right|, \tag{4}$$

where $\sum_{i,j=1}^{i,j=16} g_t(i,j)$ is the texture information in an MB of the encoded frame and $\sum_{i,j=1}^{i,j=16} G_t(i,j)$ is the texture information in the corresponding MB of the original frame.

To obtain texture information, we have applied a simple texture analysis method that uses the range value of the 3-by-3 neighborhood around the corresponding pixel to represent the pixel's texture [14]. Values of TXD and FN were obtained for each foreground MB in our entire dataset. Similar to FP versus SFD, the average of the three algorithms is shown in Figure 4(b), in which

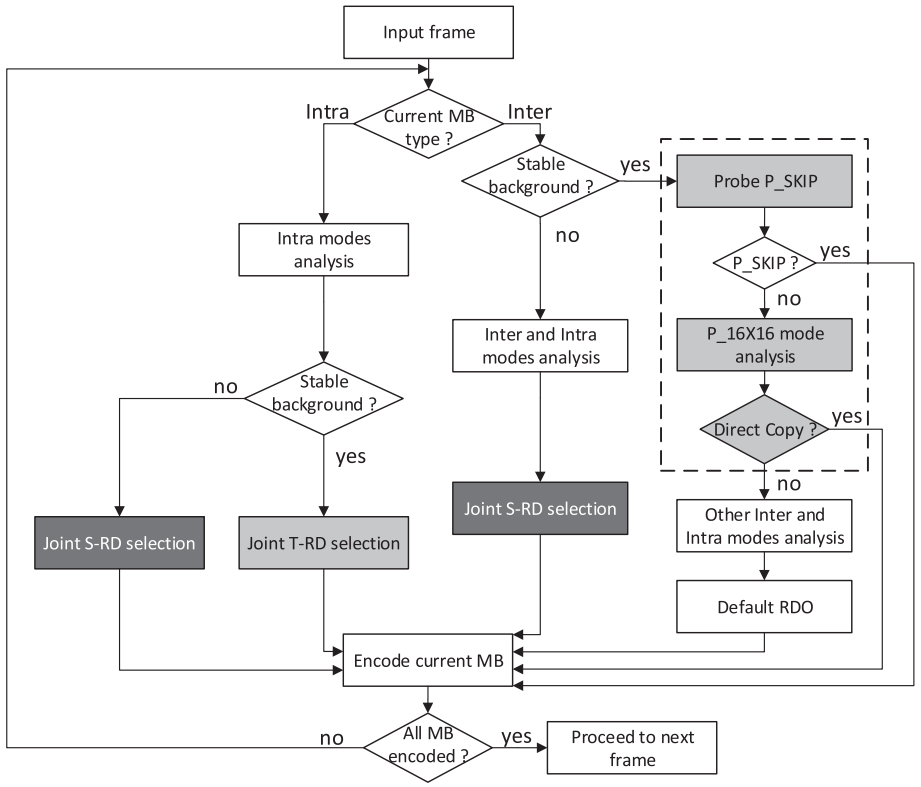


Fig. 5. Flow chart of proposed video encoding framework.

FN increases when TXD grows, and the curve looks like a quadratic function. When QP becomes larger, the curve's starting point is higher, and FN grows slower. In a small QP mode (high bit rate), the encoded video has less distortion, the overall level of FN is low, and a little texture degradation would cause a relatively large increase of FN, whereas in a large QP mode (low bit rate), the encoded video has higher distortion, the overall degree of FN is high, and the growth ratio of FN is slow.

4 PROPOSED VIDEO ENCODING FRAMEWORK

To obtain better performance of object detection on compressed videos, we propose an efficient video encoding framework for distributed wireless surveillance systems. This framework includes a *Temporal-Fluctuation-Reduced video Encoding* scheme (TFRE) for the encoding of stable background areas and a *Spatial-Texture-Preserved video Encoding* scheme (STPE) for the encoding of dynamic foreground areas. Both schemes are designed to comply with the hybrid block-based video encoding architecture, in which new mode decision and Rate Distortion Optimization (RDO) strategies are applied for intra- and interframes. The current implementation of this framework is based on the H.264/AVC standard. We consider the case that a coarse-grained classification of dynamic foreground and stable background MBs is obtained by a simple frame-differencing-based method at the encoder. The entire process for the proposed video encoding framework is illustrated in the flow chart in Figure 5, which includes two main branches, intra- and inter-MB encoding processes, for encoding the current MB of an input frame. Depending on whether the current MB is a

stable background block or a dynamic foreground block, the TFRE scheme (gray color blocks) or the STPE scheme (black color blocks) is applied.

4.1 Temporal-Fluctuation-Reduced Video Encoding Scheme

The TFRE scheme is designed to encode stable background areas to suppress unnecessary temporal fluctuation in these areas. For stable background MBs in intraframes, during the RDO process for deciding type mode and prediction mode, SFD is calculated and jointly optimized with the RDO cost. For the interframe analysis process, new strategies are introduced in the analysis of P_SKIP- and P_16×16-type modes, which are highlighted in the dashed box in Figure 5, with the objective to reduce temporal fluctuation for interblocks while maintaining acceptable distortion.

4.1.1 Intraframe Coding/Mode Selection. The intraframe RDO process of H.264/AVC consists of two steps: type mode decision from I_16×16, I_8×8, I_4×4, and I_PCM based on RDO cost, and then prediction mode decision from nine prediction options, such as vertical prediction, horizontal prediction, and so forth, based on RDO cost. And the RDO cost C is calculated by

$$C = D + \lambda \times R, \tag{5}$$

where D denotes the distortion of a candidate encoding option, R denotes the total bits of this option, and λ is the Lagrange multiplier that controls the tradeoff of rate and distortion.

We formulate a joint Temporal-fluctuation and RD (joint T-RD) mode selection problem as follows:

$$\begin{aligned} \text{Given:} \quad & \{M_i, C_i, SFD_i\} \\ \text{Find:} \quad & M^* \\ \text{Minimize:} \quad & C \\ \text{Subject to:} \quad & SFD_i \le SFD_{th}, \end{aligned} \tag{6}$$

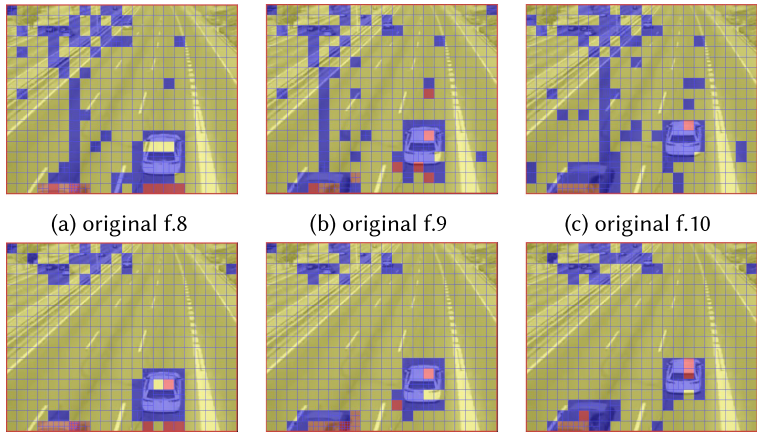
where M_i denotes the ith available type mode or prediction mode, C_i is the corresponding RDO cost, and SFD_i is the SFD value of mode i. The problem seeks to minimize the RDO cost C from a set of available modes that satisfy the SFD constraint $SFD_i \le SFD_{th}$. SFD_{th} is the N_{top}-th SFD in the ascending-order sorted array of SFD_i, and N_{top} is given by

$$N_{top} = \lceil N \times P_{top} \rceil, \tag{7}$$

where N is the total number of available modes, and P_{top} is a custom parameter that stands for the percentage of total available modes will be considered in joint T-RD selection.

Algorithm 1 is designed to solve this problem for both type mode and prediction mode selection. For a stable background MB, first, all available type modes are tried, the corresponding RDO costs and SFD values are recorded (lines 2–5 in Algorithm 1), and then the best type mode is determined based on the SFD threshold (lines 6–9); second, all available prediction modes of the selected type mode are tried, the corresponding RDO costs and SFD values are recorded (lines 10–13), and then the best prediction mode is determined based on the SFD threshold (lines 14–17). We take the x264 encoder [25] as our reference encoder. From the above description, the complexity of our proposed Algorithm 1 can stay comparable with the corresponding algorithm in the reference encoder, since the extra computations related with SFD are embedded in the original loops of the reference encoder.

4.1.2 Interframe Coding/Mode Selection. A typical interframe analysis process includes three steps: (1) Probe P_SKIP mode—that is, encode the current MB assuming no encoding residuals and no Motion Vector (MV) difference, and use only the predictive MV. The *decimate score* is computed, which indicates whether we could set the DCT coefficients to 0 given the DCT coefficients after the



(a) original f.8 (b) original f.9 (c) original f.10

(d) proposed f.8 (e) proposed f.9 (f) proposed f.10

Fig. 6. Fluctuation of P_SKIP distribution.

ALGORITHM 1: Intraframe Joint T-RD Selection

if *current MB belongs to stable background* **then**
  for *available type mode* M_{t_i} **do**
    encode current MB and store C_{t_i};
    calculate and store SFD_{t_i};
  end
  sort records in ascending order based on SFD value, and obtain valid number of records (N_t);
  obtain $SFD_{t_{th}}$ based on $N_{t_{top}} = \lceil N_t \times P_{top} \rceil$;
  find the minimum C_t, subject to $SFD_{t_i} \leq SFD_{t_{th}}$;
  output the corresponding M_t^* as the selected type mode;
  for *available prediction mode* M_{p_i} *of the selected type* M_t^* **do**
    encode current MB and store C_{p_i};
    calculate and store SFD_{p_i};
  end
  sort records in ascending order based on SFD value, and obtain valid number of records (N_p);
  obtain $SFD_{p_{th}}$ based on $N_{p_{top}} = \lceil N_p \times P_{top} \rceil$;
  find the minimum C_p, subject to $SFD_{p_i} \leq SFD_{p_{th}}$;
  output the corresponding M_p^* as the selected prediction mode;
end

actual encoding of this inter-MB [24]. If the decimate score of the current MB is less than 6, then the current MB can be encoded as P_SKIP and return [25]. (2) Otherwise, other intertype modes, including P_16×16, P_8×16, P_16×8, P_8×8, P_4×8, P_8×4, and P_4×4 modes, are all tried and the corresponding MVs are estimated, and also search is performed on those intramodes. (3) Run the RDO process and determine the best mode from all available modes.

However, the typical interframe analysis process can result in temporal fluctuation for stable background areas, which will reduce the accuracy of object detection. For example, three consecutive interframes (frames 8, 9, and 10) of the *GR* video clip is shown in the first row of Figure 6. In this figure, each block represents one MB unit, and yellow, blue, and red colors denote P_SKIP

ALGORITHM 2: Interframe Probe P_SKIP

Input: decimate score of current MB.
if *decimate score of current MB* < 6 **then**
    current MB is set as P_SKIP;
    return
else if *current MB belongs to stable background* **then**
    encode current MB based on predictive MV;
    calculate SSD_r and SFD_r based on the reconstructed MB;
    calculate SSD_s and SFD_s assume current MB as P_SKIP;
    if $SSD_s \leq d\_w \times SSD_r$ ***and*** $SFD_s \leq s\_w \times SFD_r$ **then**
        current MB is set as P_SKIP;
        return
    end
end

mode, other intermode, and intramode, respectively. Obviously, there is fluctuation of P_SKIP location distribution in these consecutive interframes. For an MB in the stable background area, when the intermode changes between P_SKIP and other interprediction modes in consecutive frames, there will be temporal fluctuation in the encoded frames, and such fluctuation might result in FP for object detection due to mistaking for new objects appearing.

We propose to reduce temporal fluctuation in interframes by designing new criteria in the analysis of intertype modes. Specifically, we expect to classify more MBs in stable background areas as P_SKIP or set the MVs of these MBs to zeros; meanwhile, we expect to maintain acceptable traditional distortion *SSD*, the *Sum of Squared Differences* between the intensities of an original MB and the intensities of an encoded MB. Based on the typical inter-MB analysis process, we design new schemes in the probe P_SKIP process and the analysis of P_16×16 mode.

In the probe P_SKIP process, for MBs dissatisfied with the original criterion in [24], we compare the encoding option of P_SKIP with the encoding option of using predictive MV, and if the P_SKIP option brings less SFD while maintaining acceptable SSD, the current MB will be set as P_SKIP. The detailed steps are described in Algorithm 2, where SSD_r and SFD_r are SSD and SFD of the reconstructed MB based on predictive MV, SSD_s and SFD_s are SSD and SFD of the current MB assuming P_SKIP encoding, and d_w and s_w are weight variables that can be customized by encoders. Compared with the x264 reference encoder, Algorithm 2 is additional; however, the overall computational complexity of interframe coding/mode selection can be reduced because more MBs can be set as P_SKIP that do not need any other intertype modes analysis and RDO.

Furthermore, for the analysis of P_16×16 mode, we design an interframe *P_16×16 Direct Copy* mode: *direct copy from the corresponding MB in the previous frame due to negligible motion in the stable background area*. If the distortion brought by assuming no motion is comparable with the distortion of reconstructed MB after motion estimation, the process will skip other intermode analyses and jump to Encode the current MB process without RDO, as shown in the flow chart in Figure 5. The detailed steps of the interframe P_16×16 Direct Copy mode are described in Algorithm 3, where SSD_{me} is the distortion of the MB based on MV_{me} after motion estimation, SSD_{dc} is the distortion of the MB based on the assumption that there is no motion and that a direct copy from the corresponding MB in the previous frame is applied, and d_w is a custom weight parameter that restricts SSD_{dc} inside a threshold of $d\_w \times SSD_{me}$. Encoding an MB in Inter P_16×16 Direct Copy mode could skip other intermode analyses and RDO, which reduces the overall computational complexity of interframe coding/mode selection.

ALGORITHM 3: Interframe P_16×16 Direct Copy Mode

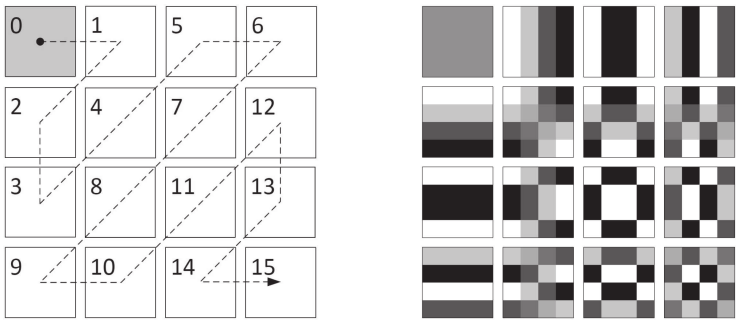
Input: MV_{me} after motion estimation in P_16×16 interanalysis.
if *current MB belongs to stable background* **then**
  encode current MB based on MV_{me};
  calculate SSD_{me} based on the reconstructed MB;
  calculate SSD_{dc} assume current MB as Direct Copy mode;
  if $SSD_{dc} \leq d\_w \times SSD_{me}$ **then**
    current MB is set as P_16×16 Direct Copy mode;
    return
  end
end

An example of the proposed intercoding scheme (combining Algorithm 2 and Algorithm 3) is shown in the second row of Figure 6. Compared with the first row, which shows results from the standard interanalysis process, after applying the proposed scheme, more background MBs are encoded as P_SKIP modes, and the distribution of P_SKIP stays stable for consecutive frames. The video snapshots from the two rows look similar, both with acceptable video quality.

4.2 Spatial-Texture-Preserved Video Encoding Scheme

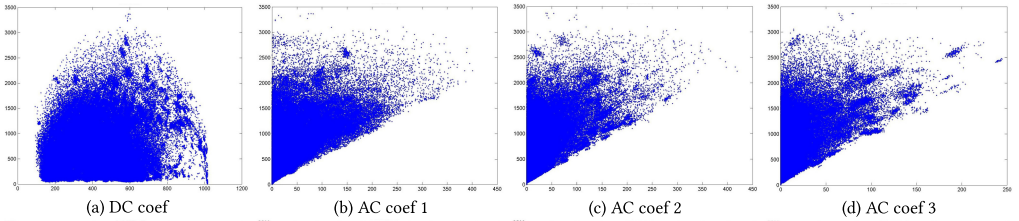
Spatial texture also plays a critical role in automatic object detection. Since 2D transform encoding, such as the Discrete Cosine Transform (DCT), is indispensable in the modern block-based hybrid video encoders, it is natural to explore the properties of spatial texture in the 2D transform domain. The texture features of an image are extracted from DCT coefficients for saliency detection in the JPEG bit-stream adaptive image retargeting applications [9]. In [10], the texture features in video saliency are detected through DCT coefficients in the MPEG4 compressed domain. Recent progress of perceptual image coding with DCT is summarized in [34], in which each image block is classified into plain, edge, or texture class based on the sum of DCT absolute coefficients. The above works are all based on 8×8 DCT in JPEG or MPEG4 but not 4×4 transform in H.264 or H.265. The features of 4×4 transform are studied in [33] with the purpose of designing a tracking-aware H.264 video compression algorithm for transportation surveillance: it has been observed that each coefficient's corresponding basis in the 4×4 transform of the H.264/AVC sharpens vertical and/or horizontal edges to varying degrees, and a new quantization table is designed that can help to identify and concentrate the compression bit rate on frequencies useful to tracking, at the cost of bit rate allocated to frequencies confusing or useless to tracking.

From the above works, we learn that the coefficients of 2D transform are highly related with spatial texture. In JPEG and MPEG4 standards, the DCT coefficients in an 8×8 block include one DC coefficient and 63 AC coefficients. Among them, the DC coefficient is the average energy over all 64 pixels in this block, and the left AC coefficients characterize the properties of the block in the frequency domain. Previous studies [9, 10, 34] show that the DCT AC coefficients can be used to represent the texture information for a block. For example, in [9], the DCT AC coefficients are classified into three parts: low-frequency (LF), medium-frequency (MF), and high-frequency (HF) parts. However, H.264/AVC uses a simplified Separable Integer 4×4 Transform (SIT) instead of 8×8 DCT [38]. The coefficients of 4×4 SIT in the H.264/AVC standard are shown in Figure 7(a), in which the first block (No. 0) with gray color denotes the DC coefficient, and the rest of the 15 blocks (No. 1–15) denote 15 AC coefficients. Figure 7(b) shows the standard basis patterns for 4×4 SIT, and the coefficients of SIT can be considered as weighting factors of a set of these basis patterns. Any image block can be reconstructed by combining the 16 basis patterns with the appropriate weight.



(a) 4×4 SIT coefficients (b) 4×4 SIT basis patterns

Fig. 7. 2D transform in H.264/AVC (4×4 SIT).



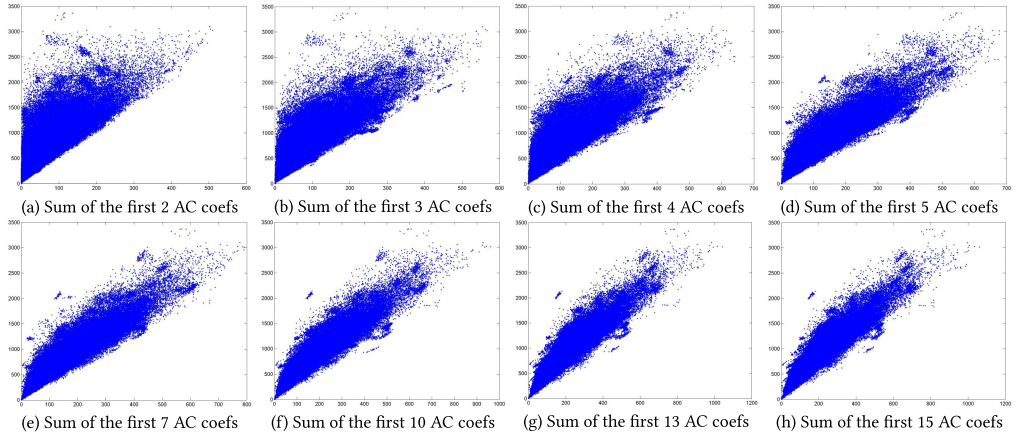
(a) DC coef (b) AC coef 1 (c) AC coef 2 (d) AC coef 3

(e) AC coef 12 (f) AC coef 13 (g) AC coef 14 (h) AC coef 15

Fig. 8. Scatter figure of SIT coefficients with spatial texture information.

First, we inspect the correlation between each coefficient of SIT with spatial texture information. The original Y channel images of *hall* are used as examples to analyze the dynamic foreground regions. All foreground MBs are applied in 4×4 SIT and the texture analysis method mentioned in Section 3. Scatter figures of the absolute value of a single SIT coefficient with texture are inspected; a DC coefficient and AC coefficients 1, 2, 3, 12, 13, 14, and 15 with spatial texture information are shown in Figure 8. We can find that the DC coefficient could not reflect the spatial texture level and that AC coefficients are related with spatial texture to a certain degree. With the number of AC coefficients increasing, the absolute value of the AC coefficient decreases generally, even close to zero (e.g., AC coefficients 13, 14, and 15), which indicates that texture details in too high frequency are in the minority. However, any single AC coefficient is not significantly correlated with spatial texture status.

We further investigate the relationship between AC coefficients of SIT and spatial texture information. We use the sum of the first x AC coefficients to represent spatial texture information, where x is an integer more than 0 and less than 16. In Figure 9, scatter figures are shown for the sum of the first 2, 3, 4, 5, 7, 10, 13, and 15 AC coefficients with spatial texture information, respectively. The scatter figure of the first AC coefficient has already been shown in Figure 8(b). We find that



(a) Sum of the first 2 AC coefs (b) Sum of the first 3 AC coefs (c) Sum of the first 4 AC coefs (d) Sum of the first 5 AC coefs

(e) Sum of the first 7 AC coefs (f) Sum of the first 10 AC coefs (g) Sum of the first 13 AC coefs (h) Sum of the first 15 AC coefs

Fig. 9. Scatter figure of sum of the first x AC coefficients with spatial texture information.

the correlation becomes more clear when more AC coefficients are considered with x increasing from 1 to 5, and as x continues to increase, there is no obvious improvement of the correlation. It is well known that different computer vision algorithms work based on different levels of features, and therefore, we prefer not to select to preserve specific frequencies; in other words, we try to protect all the frequency details as the original ecology.

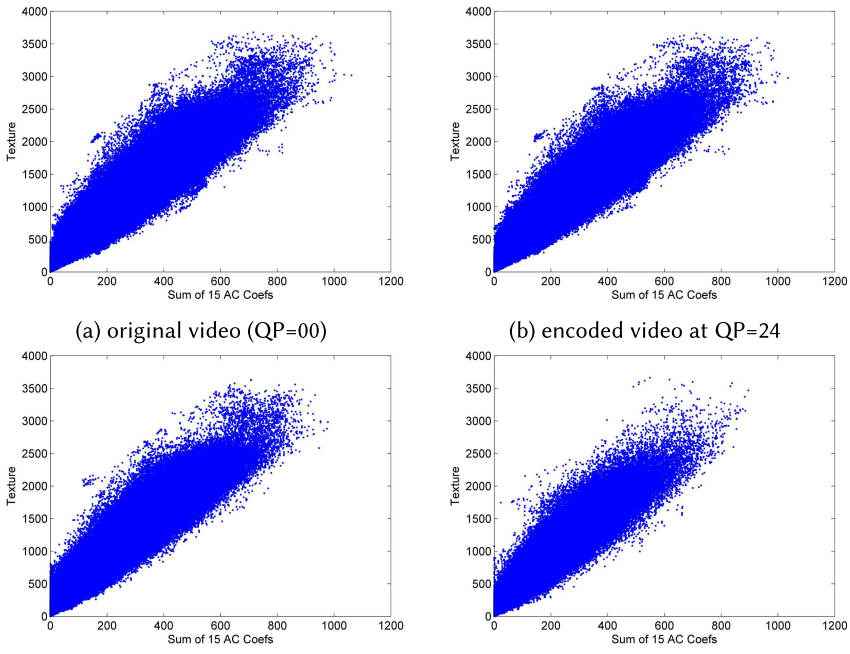
Consequently, we introduce the measure *SSAC*, *Sum-of-absolute 15 SIT AC Coefficients*, to depict the spatial texture information of a 4×4 image block:

$$SSAC = \sum_{i=1}^{i=15} |AC_i|, \tag{8}$$

where AC_i is the ith SIT AC coefficient of one 4×4 image block in an MB. We investigate the entire video dataset, which includes different compression-level videos (QP from 24 to 48, step size is 2) and the original raw videos. The scatter figures of the original raw videos (where QP is marked as 00) and the encoded video using QP 24, 36, and 48 are shown in Figure 10. Based on the scatter figures of the entire video dataset, the distribution becomes more and more concentrated and regular as the compression ratio (QP) increases. We use SSAC value 10 as intervals to average data points in the scatter figure and then obtain curves for all QP settings, which are shown in Figure 11. We can find that there is a positive linear correlation between SSAC and spatial texture, no matter how much compression is introduced.

The correlation between spatial texture and SSAC is inspected using the Linear Correlation Coefficient (LCC), the Spearman Rank Order Correlation Coefficient (SROCC), and the Kendall Rank Correlation Coefficient (KRCC), respectively. The correlation coefficients are summarized in Table 1, in which QP 00 denotes the original raw video. The results of LCC are all above 0.96 (average value 0.965), those of SROCC are all higher than 0.97 (average value 0.978), and those of KRCC are all higher than 0.87 (average value 0.879). These results indicate that there is a significant positive linear correlation between SSAC and spatial texture.

By far, we have proposed the ideal descriptor in the 2D transform domain, SSAC, to depict the spatial texture information in video encoding scenarios. Inheriting the concept of texture degradation from TXD, which is defined in Equation (4), we introduce a new TXD^{SIT} as a basic unit to



(a) original video (QP=00) (b) encoded video at QP=24

(c) encoded video at QP=36 (d) encoded video at QP=48

Fig. 10. Scatter figure of SSAC with spatial texture information.

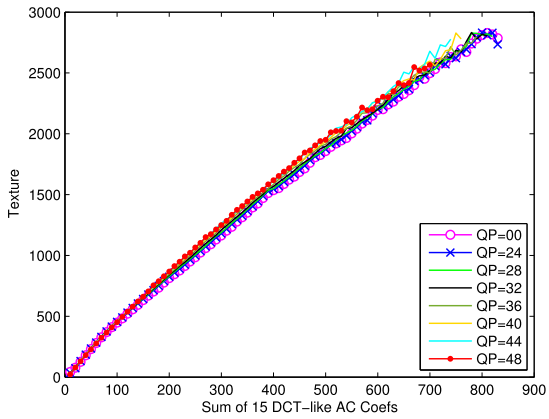


Fig. 11. The relationship between SSAC and spatial texture information.

Table 1. Correlation Coefficients between Spatial Texture Information and SSAC

QP	00	24	26	28	30	32	34	36	38	40	42	44	46	48
LCC	0.962	0.964	0.964	0.964	0.965	0.965	0.965	0.965	0.966	0.966	0.966	0.966	0.966	0.965
SROCC	0.975	0.977	0.977	0.977	0.978	0.978	0.978	0.978	0.978	0.979	0.979	0.979	0.979	0.980
KRCC	0.871	0.874	0.875	0.875	0.876	0.877	0.878	0.879	0.881	0.882	0.883	0.884	0.886	0.889

represent texture degradation in a 4×4 block, which is given by

$$TXD^{SIT} = \left| \sum_{i=1}^{i=15} |AC_i| - \sum_{i=1}^{i=15} |ac_i| \right|, \tag{9}$$

where $\sum_{i=1}^{i=15} |AC_i|$ is texture information of a 4×4 block in an MB of the original frame and $\sum_{i=1}^{i=15} |ac_i|$ is texture information of the corresponding 4×4 block in the same MB of the encoded frame. The descriptor of texture degradation TXD^{SIT} brings unique benefits in the video encoding context, including (1) convenient calculation, (2) low computational complexity, and (3) finer-grained tuning than the original TXD in MB unit.

The block-based video coding can be summarized as an intra/intertype mode (macroblock partitions) decision and an intra/interprediction (nine prediction options or motion-compensated prediction) mode decision. To protect spatial texture during the video encoding process, we formulate a joint Spatial-texture and RD (joint S-RD) mode selection problem for both intra- and interframes as follows:

$$\begin{aligned} \text{Given:} \quad & \{M_i, C_i, TXD_i^{SIT}\} \\ \text{Find:} \quad & M^* \\ \text{Minimize:} \quad & C \\ \text{Subject to:} \quad & TXD_i^{SIT} \leq TXD_{th}^{SIT}, \end{aligned} \tag{10}$$

where M_i denotes the ith available type mode or prediction mode in an intra/interframe, C_i is the corresponding RDO cost, and TXD_i^{SIT} is the TXD^{SIT} value of mode i. The problem seeks to minimize the RDO cost C from a set of available modes that satisfy the TXD^{SIT} constraint $TXD_i^{SIT} \leq TXD_{th}^{SIT}$. TXD_{th}^{SIT} is the N_{top}-th TXD^{SIT} in the ascending-order sorted array of TXD_i^{SIT}, and N_{top} is given by

$$N_{top} = \lceil N \times P_{top} \rceil, \tag{11}$$

where N is the total number of available modes, and P_{top} is a custom parameter that stands for the percentage of total available modes will be considered in joint S-RD selection.

Algorithm 4 is designed to solve this problem for both intra/inter-type mode and intra/interprediction mode selection. For a dynamic foreground MB, first, the SSAC of the original video is calculated and stored as a benchmark, and then all the available intra/intertype modes are tried, the corresponding RDO costs and TXD^{SIT} values are recorded (lines 4–7 in Algorithm 4), and the best type mode is determined based on the TXD^{SIT} threshold (lines 8–11); second, all available intra/interprediction modes of the selected type mode are tried, the corresponding RDO costs and TXD^{SIT} values are recorded (lines 12–15), and then the best prediction mode is determined based on the TXD^{SIT} threshold (lines 16–19). Based on the above procedure, our proposed Algorithm 4 can maintain the same level as the x264 reference encoder in the computational complexity, since the computing related with TXD does not bring extra loops to the reference encoder.

5 PERFORMANCE EVALUATION

We evaluate the proposed video encoding framework by applying object detection algorithms on a variety of compressed videos. The eight raw videos shown in Figure 3 and a new video dataset from PETS 2017 datasets [27] are used for this test. There are uniform resolutions (352×288), frame rates (25 fps), and durations (12 sec) in the dataset 1. Dataset 2 contains the ARENA dataset (AD), which includes four nonoverlapping fields of view at each corner of a truck outdoors, and the maritime IPATCH dataset (ID), which includes one view at stern and three starboard views (sv) of one ship. Different from dataset 1, eight videos in dataset 2 cover higher resolutions (1280×960), higher frame rates (30 fps), and longer durations (20∼100 sec), and more details can be found in

Table 2. Video Information for Dataset 2

Video Name	AD Right	AD Left	AD Ahead	AD Behind	ID Stern	ID sv 1	ID sv 2	ID sv 3
SI index	83.71	91.78	95.62	123.63	75.69	53.37	47.08	59.64
TI index	21.14	61.45	42.73	47.37	10.47	12.86	16.99	8.86
Length (sec)	20	60	40	40	60	100	70	90

Table 3. Video Compression Parameters

GOP Structure	IPPP	GOP Size	20
Rate control	Constant QP	QP range	28-46
Intra/inter custom parameters	P_{top}	d_w	s_w
	0.1	6	0.1

ALGORITHM 4: Intra/Inter-MB Joint S-RD Selection

Input: Intraprediction options or intermotion vectors based on SATD scores
if *current MB belongs to dynamic foreground* **then**
  4×4 SIT on the original video, and store the original SSAC;
  for *available type mode* M_{t_i} *of intra/inter-MB* **do**
    encode current MB based on intra/interprediction, store C_{t_i} and the corresponding SSAC;
    calculate and store $TXD_{t_i}^{SIT}$;
  end
  sort records in ascending order based on TXD^{SIT} value, and obtain valid number of records (N_t);
  obtain $TXD_{t_{th}}^{SIT}$ based on $N_{t_{top}} = \lceil N_t \times P_{top} \rceil$;
  find the minimum C_t, subject to $TXD_{t_i}^{SIT} \leq TXD_{t_{th}}^{SIT}$;
  select the corresponding M_t^* as the optimal type mode;
  for *available prediction mode* M_{p_i} *based on the selected type* M_t^* **do**
    encode current MB or store C_{p_i} and the corresponding SSAC;
    calculate and store $TXD_{p_i}^{SIT}$;
  end
  sort records in ascending order based on TXD^{SIT} value, and obtain valid number of records (N_p);
  obtain $TXD_{p_{th}}^{SIT}$ based on $N_{p_{top}} = \lceil N_p \times P_{top} \rceil$;
  find the minimum C_p, subject to $TXD_{p_i}^{SIT} \leq TXD_{p_{th}}^{SIT}$;
  output the corresponding M_p^* of the selected type M_t^* as the optimal mode;
end

Table 2. The Spatial Information (SI) index and Temporal Information (TI) index of a sequence, which are defined by ITU-T P.910 [29] and are directly related to video compression complexity, are also included in Table 2. The x264 encoder (version 0.142.x) is configured to encode videos using one-pass mode with medium speed, and the compression settings are summarized in Table 3. The aforementioned three object detection algorithms (GMM, GMG, and ABL) are applied on these compressed videos. One motivation to include relatively higher QP values in our tests is that medium and high compression ratios are used in many wide-area, large-scale, or sparse wireless camera networks with limited bandwidth and energy constraints. For example, a

wide-area and large-scale camera network is implemented in [21], a long-duration and large-scale environmental monitoring application is introduced in [7], the deployment of sparse sensor networks in large areas is studied in [8], and the deployment of airborne camera networks is introduced in [30]. These practical systems operate in a bandwidth range of 40kbps to 300kbps, providing video observations with around 0.01 to 0.1 bits per pixel (BPP). The QP values in our experiments could produce videos with bandwidth and BPP ranges consistent with these practical wireless camera systems. Moreover, a similar QP range (28–44) was adopted by other researchers for studies on subjective video quality in [26] and [23], which demonstrated that the perceptual quality of videos encoded with medium and high QP is acceptable.

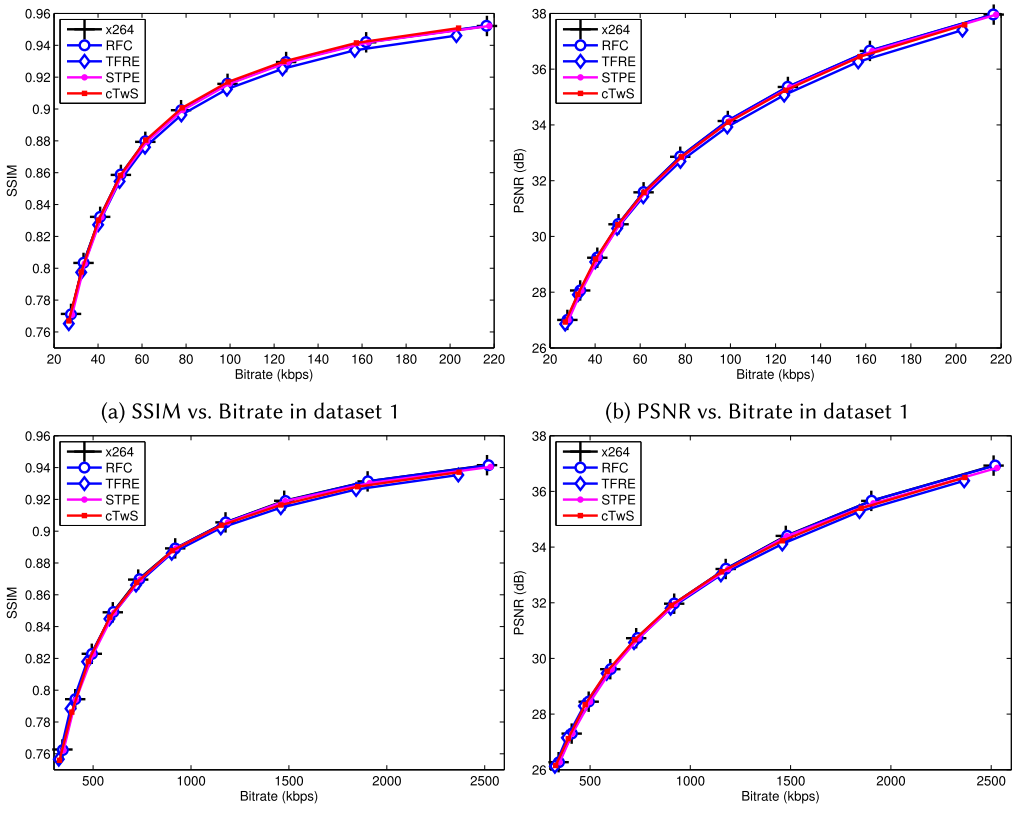
We evaluate the performance of the proposed algorithms in terms of both pixel-level detection accuracy and object-level detection accuracy on the two datasets. Evaluation of detection at the object level is straightforward, while more precise detection at the pixel level provides more insight into strengths and weaknesses of detection performance [3, 11], based on which solutions could be designed to improve object detection performance.

5.1 Evaluation of the Proposed Algorithms in Pixel Level

The performances of the proposed TRFE scheme, STPE scheme, and combined TFRE with STPE scheme (short for cTwS) are compared to the H.264/AVC-based open-source encoder x264 and the Reducing Flicker video Coding approach (RFC) [6]. The objective of RFC is to improve perceptual video quality by reducing flicker effects, and it considers the distortions not only between the encoded video and the original video but also in the temporal domain in the encoded video during the intrarate distortion optimization process.

First, we compare the objective video quality and the corresponding bitrate of the five schemes in dataset 1 and dataset 2. The industrial standard PSNR and Structural Similarity (SSIM) are applied to the compressed videos. We evaluate the average performance of the eight different video sequences in dataset 1 and dataset 2 separately at the same QP. The resulting PSNR and SSIM with the corresponding bit rates are shown in Figure 12. The R-D performances of RFC are nearly identical with those of x264 in every QP for both datasets. The curves of STPE almost overlap completely with the ones of x264 for both SSIM and PSNR, which indicates that the proposed STPE scheme has little impact on the Rate Distortion performance. The PSNR and SSIM values of the TFRE scheme decrease slightly, whereas the bitrate is saved compared with ones of x264 encoding. The slight decrease in bitrate is due to the fact that TFRE encodes more inter-MBs in P_SKIP modes. For the combined cTwS scheme, compared with x264 encoding: in dataset 1, the PSNR and SSIM values of cTwS decrease slightly by 0.102dB and 0.001 on average, respectively, whereas cTwS brings down the bitrate by 2.04kbps on average; in high-resolution videos of dataset 2, its PSNR and SSIM decrease by 0.157dB and 0.004 on average, respectively, whereas it saves the bitrate by 31.67kbps on average. Overall, the Rate Distortion performances of the proposed algorithms are comparable with those of the x264 encoder.

Next, we evaluate the overall performance of object detection in pixel level through F_1 scores. The average F_1 scores of the eight videos in each dataset for the three object detection algorithms are shown in Figure 13. Though the three object detection algorithms have different ranges of F_1 in two datasets, the detection performance degrades when QP increases. The curves of the RFC scheme always nearly overlap with those of x264 except for negligible improvements of the ABL algorithm. The performance gains of the STPE scheme are larger than ones of RFC and distributed evenly over different QPs. The benefits of the TFRE scheme upon x264 for three algorithms on both datasets are noticeable, and the gain of TFRE is higher with larger QP values. The cTwS scheme results in the largest F_1 scores for different QP values in every algorithm on both datasets. More



(a) SSIM vs. Bitrate in dataset 1

(b) PSNR vs. Bitrate in dataset 1

(c) SSIM vs. Bitrate in dataset 2

(d) PSNR vs. Bitrate in dataset 2

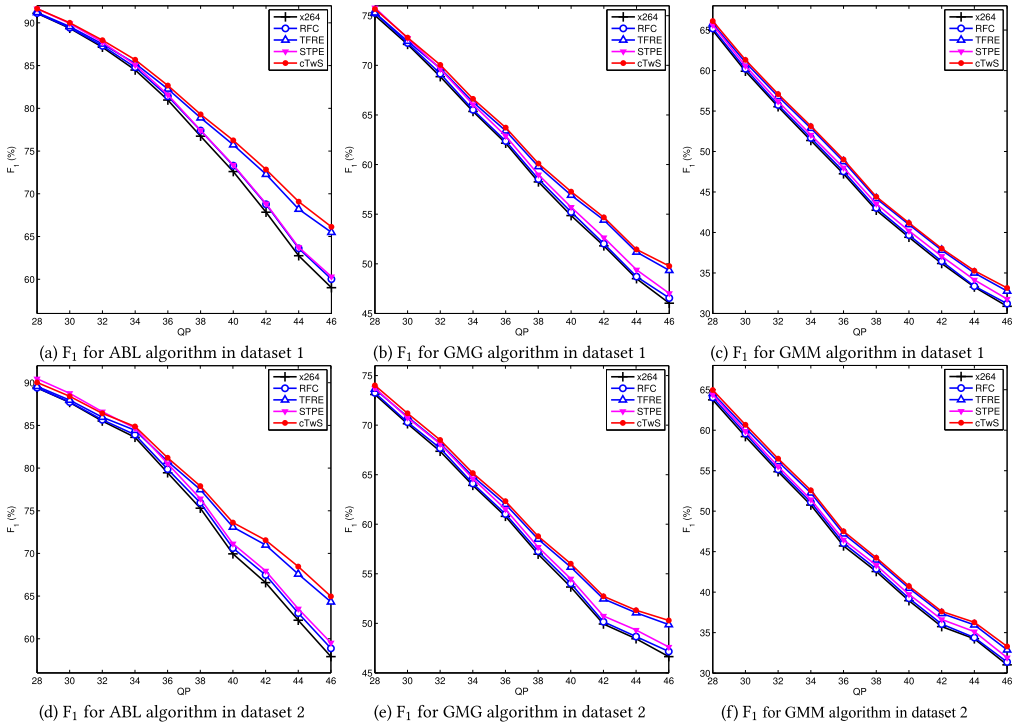
Fig. 12. Rate-distortion curves of proposed schemes.

specifically, there are noticeable gains of cTwS over x264 for ABL (average 2.96% and 2.99% for two datasets), and modest gains for GMG (1.92% and 1.94%) and GMM (1.72% and 1.76%).

Finally, we summarize the average F_1 scores of the three object detection algorithms on two datasets in Table 4. The numbers in the Δ rows denote the gains of cTwS over the x264 encoder. Three points could be reached based on the summary table and above figures: (1) both the R-D and the object detection performances of RFC are nearly identical with that of x264, (2) the R-D performance of cTwS is comparable to that of the x264 encoder and RFC, and (3) the improvement on detection performance of cTwS at every QPs is obvious, and the gain of cTwS is higher with larger QP values. These results indicate that, by reducing temporal fluctuation in stable background areas and preserving spatial texture in foreground areas, the proposed video encoding framework could effectively improve the accuracy of object detection in pixel level for different types of detection algorithms with no impact on the R-D performance.

5.2 Evaluation of the Proposed Algorithms in Object Level

To evaluate the performance of the proposed algorithms in object level, a uniform postprocessing procedure is adopted to the pixel-level results of three detection algorithms. The postprocessing modules includes the following:



(a) F_1 for ABL algorithm in dataset 1 (b) F_1 for GMG algorithm in dataset 1 (c) F_1 for GMM algorithm in dataset 1

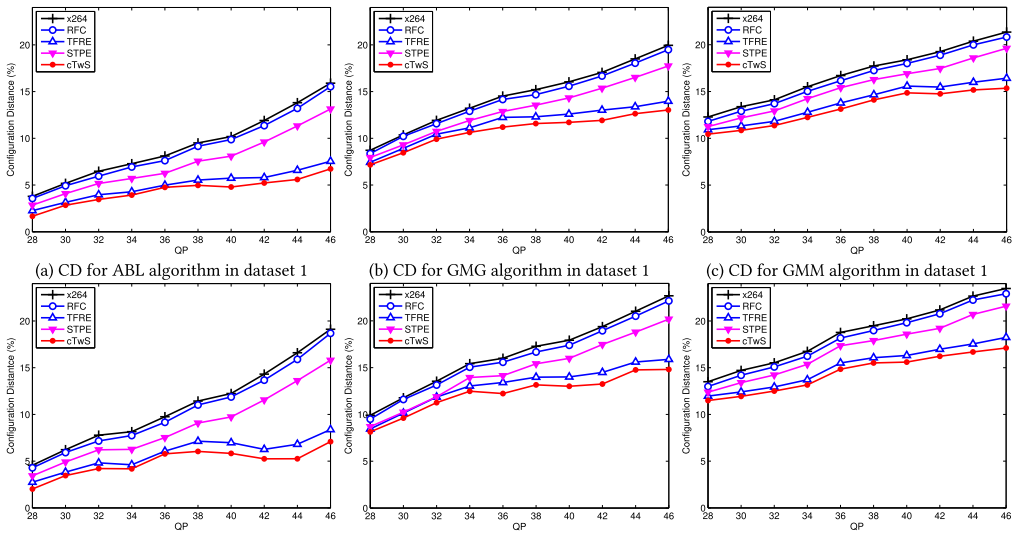
(d) F_1 for ABL algorithm in dataset 2 (e) F_1 for GMG algorithm in dataset 2 (f) F_1 for GMM algorithm in dataset 2

Fig. 13. F_1 scores of test videos in proposed schemes.

Table 4. Average Results of Proposed Algorithms in Pixel Level

QP		28	30	32	34	36	38	40	42	44	46
Dataset 1 F_1 (%)	x264	77.03	73.80	70.51	67.06	63.45	59.25	55.62	51.94	48.17	45.31
	RFC	77.21	73.97	70.74	67.33	63.77	59.65	56.06	52.42	48.56	45.92
	TFRE	77.70	74.42	71.19	67.72	64.16	59.98	56.43	52.85	49.10	46.37
	STPE	77.43	74.34	71.34	68.14	64.81	60.96	57.88	54.83	51.46	49.20
	cTwS	77.81	74.71	71.70	68.48	65.13	61.27	58.23	55.18	51.94	49.68
	Δ	0.78	0.91	1.19	1.42	1.68	2.02	2.61	3.24	3.77	4.37
Dataset 2 F_1 (%)	x264	75.41	72.35	69.25	66.09	61.98	58.27	54.19	50.75	48.28	45.18
	RFC	75.57	72.54	69.49	66.38	62.32	58.69	54.63	51.23	48.68	45.80
	TFRE	75.92	73.03	70.08	67.17	63.33	59.98	56.42	53.60	51.53	49.02
	STPE	76.21	73.13	70.08	66.91	62.83	59.14	55.12	51.77	49.32	46.35
	cTwS	76.34	73.43	70.47	67.54	63.68	60.31	56.79	53.97	52.02	49.52
	Δ	0.93	1.08	1.22	1.45	1.70	2.04	2.60	3.22	3.74	4.34

(1) Median filtering (5×5 rectangular aperture)
(2) Morphological operations (first opening then closing with 3×3 square structure)
(3) Connected-component labeling (eight-way connectivity)
(4) Region thresholding (240 pixels)



(a) CD for ABL algorithm in dataset 1 (b) CD for GMG algorithm in dataset 1 (c) CD for GMM algorithm in dataset 1

(d) CD for ABL algorithm in dataset 2 (e) CD for GMG algorithm in dataset 2 (f) CD for GMM algorithm in dataset 2

Fig. 14. Configuration Distance (CD) of testing videos in proposed schemes.

For the object-level detection accuracy, we calculate the Configuration Distance (CD) [2], which measures the difference between the amount of GT objects and AR objects according to their presence. For one given frame, the CD_f can be calculated by

$$CD_f = \left| \frac{AR_o - GT_o}{max(GT_o, 1)} \right|, \tag{12}$$

where AR_o and GT_o are the numbers of AR objects and GT objects in the frame. The CD of a video sequence is obtained by the average of CD_f in each frame. In our experiments, object detection results from the raw videos are regarded as GT, and results from the compressed videos are AR. Ideally, if a video is compressed in a lossless way, the corresponding CD is 0. Lossy compression inevitably degrades the performance of object detection, and both false positives and false negatives (i.e., detection mistaking and detection missing) could result in an increase of CD.

The performance of proposed algorithms in object level through CD value in two datasets is shown in Figure 14. Despite different ranges of CD values in two datasets, the trends of the three detection algorithms are similar. The curves of RFC are always very close to the ones of x264 for three algorithms. The minimum gains of STPE over x264 for ABL, GMG, and GMM algorithms (−0.94%, −0.77%, and −1.03% in dataset 1; −1.13%, −1.21%, and −1.12% in dataset 2) are larger than the maximum gains of RFC (−0.60%, −0.53%, and −0.56% in dataset 1; −0.71%, −0.61%, and −0.60% in dataset 2). The TFRE scheme improves the detection performance significantly. The combined cTwS scheme attains the remarkable improvement (average gains: −4.82%, −3.73%, and 3.68% in dataset 1; −6.10%, −4.22%, and −4.12% in dataset 2, respectively). The average CD values of the three object detection algorithms on two datasets are also summarized in Table 5. The values in the Δ rows denote the gains of cTwS over the x264 encoding. In summary, the average gains of RFC over x264 are quite limited (average −0.42% in dataset 1 and −0.47% in dataset 2, respectively), STPE achieves better improvement than RFC (with average gains −1.60% and −1.86%), the improvement of TFRE is considerable (with average gains −3.42% and −4.04%), and cTwS achieves the maximum average gains (−4.08% and −4.82%).

Table 5. Average Results of Proposed Algorithms in Object Level

QP		28	30	32	34	36	38	40	42	44	46
Dataset 1 CD (%)	x264	8.26	9.64	10.84	12.00	13.11	14.15	14.87	16.07	17.56	19.06
	RFC	7.91	9.34	10.41	11.62	12.64	13.69	14.48	15.64	17.08	18.61
	TFRE	7.35	8.52	9.61	10.61	11.51	12.46	13.10	14.14	15.48	16.83
	STPE	6.87	7.80	8.74	9.40	10.34	10.83	11.30	11.43	11.98	12.65
	cTwS	6.42	7.38	8.24	8.93	9.69	10.22	10.45	10.63	11.13	11.70
	Δ	−1.84	−2.26	−2.60	−3.07	−3.42	−3.93	−4.42	−5.44	−6.43	−7.36
Dataset 2 CD (%)	x264	9.32	10.91	12.28	13.44	14.84	16.07	16.80	18.29	20.09	21.75
	RFC	8.93	10.57	11.79	13.01	14.30	15.55	16.36	17.79	19.54	21.24
	TFRE	8.17	9.52	10.77	11.85	13.00	14.13	14.77	16.08	17.70	19.19
	STPE	7.72	8.79	9.87	10.45	11.66	12.39	12.43	12.57	13.32	14.17
	cTwS	7.22	8.33	9.32	9.94	10.95	11.57	11.48	11.57	12.23	13.00
	Δ	−2.10	−2.58	−2.96	−3.50	−3.89	−4.50	−5.32	−6.72	−7.86	−8.75

Table 6. Computational Complexity of Algorithms

Algorithms	x264	RFC	TFRE	STPE	cTwS
Dataset 1 complexity (ms)	1,934.985	2,131.977	1,278.990	2,021.941	1,309.986
Gains (%)	—	+10.18	−33.90	+4.49	−32.30
Dataset 2 Complexity (ms)	18,743.861	20,622.032	14,814.014	19,655.053	15,391.520
Gains (%)	—	+10.02	−20.97	+4.86	−17.89

5.3 Evaluation of the Computational Complexity

The computational complexity of algorithms is a crucial design factor for distributed wireless surveillance systems. All video encoding in this article was performed exclusively on a computer based on an Intel Xeon E5-2637 v3 (3.50GHz) processor running on a Windows 7 Enterprise operating system. Computational complexity was measured by the encoding time for the x264 encoder, RFC approach, TFRE scheme, STPE scheme, and combined TFRE with STPE scheme (cTwS). Computational complexity is evaluated by the average encoding time of running separately three times for both the 80 test cases in dataset 1 and the 80 test cases in dataset 2, which is summarized in Table 6. Each dataset consists of eight different videos in 10 different QP configurations.

The computational complexity of the x264 encoder in Table 6 is regarded as a benchmark. The RFC approach increases more than 10% in complexity due to extra computing introduced during the intra-RDO process. The proposed TFRE scheme reduces computational complexity significantly (−33.90% and −20.97%, respectively) thanks to avoiding unnecessary Inter mode analyses and RDO processes. The proposed STPE scheme maintains comparable complexity (less than 5%). Finally, the combined TFRE with STPE scheme achieves −32.30% and −17.89% reductions in computational complexity for dataset 1 and dataset 2, respectively. The reduction in complexity will provide considerable benefits for distributed wireless surveillance applications.

6 CONCLUSION

In this article, we have proposed an efficient video encoding framework that aims at improving the performance of object detection on compressed videos in distributed wireless surveillance systems. This framework includes the *Temporal-Fluctuation-Reduced video Encoding* scheme (TFRE) for the encoding of stable background areas and the *Spatial-Texture-Preserved video Encoding* scheme (STPE) for the encoding of dynamic foreground areas. Besides, this framework is compliant with the block-based hybrid encoding architecture, and its computational complexity of H.264-based implementation is reduced significantly to that of common H.264 encoding schemes. Experimental results on a variety of encoder settings and object detection algorithms indicate that, compared with traditional encoding schemes, the framework improves the accuracy of object detection and results in lower bitrate and significantly reduced complexity with comparable video quality in terms of PSNR and SSIM. This standard-compliant video encoding framework can promote the development and applications of many distributed wireless surveillance systems. In the future, we plan to implement the proposed encoding framework in the newly developed HEVC standard.

REFERENCES

[1] Andrew D. Bagdanov, Marco Bertini, Alberto Del Bimbo, and Lorenzo Seidenari. 2011. Adaptive video compression for video surveillance applications. In *2011 IEEE International Symposium on Multimedia (ISM'11)*. IEEE, 190–197.

[2] Axel Baumann, Marco Boltz, Julia Ebling, Matthias Koenig, Hartmut S. Loos, Marcel Merkel, Wolfgang Niem, Jan Karl Warzelhan, and Jie Yu. 2008. A review and comparison of measures for automatic video surveillance systems. *EURASIP Journal on Image and Video Processing* 1 (2008), 824726.

[3] Sebastian Brutzer, Benjamin Höferlin, and Gunther Heidemann. 2011. Evaluation of background subtraction techniques for video surveillance. In *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11)*. IEEE, 1937–1944.

[4] Jianshu Chao, Robert Huitl, Eckehard Steinbach, and Damien Schroeder. 2015. A novel rate control framework for SIFT/SURF feature preservation in H. 264/AVC video compression. *IEEE Transactions on Circuits and Systems for Video Technology* 25, 6 (2015), 958–972.

[5] Xiang Chen, Jenq-Neng Hwang, Kuan-Hui Lee, and Ricardo L. de Queiroz. 2015. Quality-of-content (QoC)-driven rate allocation for video analysis in mobile surveillance networks. In *2015 IEEE 17th International Workshop on Multimedia Signal Processing (MMSP'15)*. IEEE, 1–6.

[6] Seong Soo Chun, Jung-Rim Kim, and Sanghoon Sull. 2006. Intra prediction mode selection for flicker reduction in H. 264/AVC. *IEEE Transactions on Consumer Electronics* 52, 4 (2006), 1303–1310.

[7] Peter Corke, Tim Wark, Raja Jurdak, Wen Hu, Philip Valencia, and Darren Moore. 2010. Environmental wireless sensor networks. *Proceedings of IEEE* 98, 11 (2010), 1903–1917.

[8] Wan Du, Zhenjiang Li, Jansen Christian Liando, and Mo Li. 2016. From rateless to distanceless: Enabling sparse sensor network deployment in large areas. *IEEE/ACM Transactions on Networking* 24, 4 (2016), 2498–2511.

[9] Yuming Fang, Zhenzhong Chen, Weisi Lin, and Chia-Wen Lin. 2012. Saliency detection in the compressed domain for adaptive image retargeting. *IEEE Transactions on Image Processing* 21, 9 (2012), 3888–3901.

[10] Yuming Fang, Weisi Lin, Zhenzhong Chen, Chia-Ming Tsai, and Chia-Wen Lin. 2014. A video saliency detection model in compressed domain. *IEEE Transactions on Circuits and Systems for Video Technology* 24, 1 (2014), 27–38.

[11] Nil Goyette, Pierre-Marc Jodoin, Fatih Porikli, Janusz Konrad, and Prakash Ishwar. 2012. Changedetection.net: A new change detection benchmark dataset. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW'12)*. IEEE, 1–8.

[12] Hai-Miao Hu, Bo Li, Weiyao Lin, Wei Li, and Ming-Ting Sun. 2012. Region-based rate control for H. 264/AVC for low bit-rate applications. *IEEE Transactions on Circuits and Systems for Video Technology* 22, 11 (2012), 1564–1576.

[13] Weiming Hu, Tieniu Tan, Liang Wang, and Steve Maybank. 2004. A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* 34, 3 (2004), 334–352.

[14] MathWorks Inc. 2006. Local range of image-MATLAB rangefilt. Retrieved March 21, 2017, from http://www.mathworks.com/help/images/ref/rangefilt.html.

[15] Amaya Jiménez-Moreno, Eduardo Martinez-Enriquez, Vipin Kumar, and Fernando Díaz-de María. 2014. Standard-compliant low-pass temporal filter to reduce the perceived flicker artifact. *IEEE Transactions on Multimedia* 16, 7 (2014), 1863–1873.

[16] Emmanouil Kafetzakis, Christos Xilouris, Michail Alexandros Kourtis, Marcos Nieto, Iveel Jargalsaikhan, and Suzanne Little. 2013. The impact of video transcoding parameters on event detection for surveillance systems. In *2013 IEEE International Symposium on Multimedia (ISM'13)*. IEEE, 333–338.

[17] Lingchao Kong and Rui Dai. 2016. Temporal-fluctuation-reduced video encoding for object detection in wireless surveillance systems. In *2016 IEEE International Symposium on Multimedia (ISM'16)*. IEEE, 126–132.

[18] Lingchao Kong and Rui Dai. 2017. Object-detection-based video compression for wireless surveillance systems. *IEEE MultiMedia* 24, 2 (2017), 76–85.

[19] Lingchao Kong, Rui Dai, and Yuchi Zhang. 2016. A new quality model for object detection using compressed videos. In *2016 IEEE International Conference on Image Processing (ICIP'16)*. IEEE, 3797–3801.

[20] Pavel Korshunov and Wei Tsang Ooi. 2011. Video quality for face detection, recognition, and tracking. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 7, 3 (2011), 14.

[21] Thomas Kuo, Zefeng Ni, Carter De Leo, and B. S. Manjunath. 2010. Design and implementation of a wide area, large-scale camera network. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW'10)*. IEEE, 25–32.

[22] Mikołaj Leszczuk. 2014. Optimising task-based video quality. *Multimedia Tools and Applications* 68, 1 (2014), 41–58.

[23] Zhan Ma, Meng Xu, Yen-Fu Ou, and Yao Wang. 2012. Modeling of rate and perceptual quality of compressed video as functions of frame rate and quantization stepsize and its applications. *IEEE Transactions on Circuits and Systems for Video Technology* 22, 5 (2012), 671–682.

[24] Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG. 2001. Working Draft Number 2, Revision 0 (WD-2). *JVT-B118*.

[25] VideoLAN Organization. 2005. x264, the best H.264/AVC encoder. Retrieved March 21, 2017, from http://www.videolan.org/developers/x264.html.

[26] Yen-Fu Ou, Zhan Ma, Tao Liu, and Yao Wang. 2011. Perceptual quality assessment of video considering both frame rate and quantization artifacts. *IEEE Transactions on Circuits and Systems for Video Technology* 21, 3 (2011), 286–298.

[27] Luis Patino, Tahir Nawaz, Tom Cane, and James Ferryman. 2017. PETS 2017: Dataset and challenge. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17) Workshops*.

[28] R. M. T. P. Rajakaruna, W. A. C. Fernando, and J. Calic. 2011. Application-aware video coding architecture using camera and object motion-models. In *2011 6th IEEE International Conference on Industrial and Information Systems (ICIIS'11)*. IEEE, 76–81.

[29] ITU-T Recommendation. 2008. P.910. *Subjective Video Quality Assessment Methods for Multimedia Applications*, 910–200804.

[30] Danileno Rosário, José Arnaldo Filho, Denis Rosário, Aldri Santosy, and Mário Gerla. 2017. A relay placement mechanism based on UAV mobility for satisfactory video transmissions. In *2017 16th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net'17)*. IEEE, 1–8.

[31] Lauro Snidaro, Ingrid Visentini, and Gian Luca Foresti. 2012. Fusing multiple video sensors for surveillance. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 8, 1 (2012), 7.

[32] Andrews Sobral and Antoine Vacavant. 2014. A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. *Computer Vision and Image Understanding* 122 (2014), 4–21.

[33] Eren Soyak, Sotirios Tsaftaris, and Aggelos K. Katsaggelos. 2011. Low-complexity tracking-aware H. 264 video compression for transportation surveillance. *IEEE Transactions on Circuits and Systems for Video Technology* 21, 10 (2011), 1378–1389.

[34] Ee-Leng Tan and Woon-Seng Gan. 2015. Perceptual image coding with discrete cosine transform. In *Perceptual Image Coding with Discrete Cosine Transform*. Springer, 21–41.

[35] Bulent Tavli, Kemal Bicakci, Ruken Zilan, and Jose M. Barcelo-Ordinas. 2012. A survey of visual sensor network platforms. *Multimedia Tools and Applications* 60, 3 (2012), 689–726.

[36] Peng Wang, Yongfei Zhang, Hai-Miao Hu, and Bo Li. 2013. Region-classification-based rate control for flicker suppression of I-frames in HEVC. In *2013 20th IEEE International Conference on Image Processing (ICIP'13)*. IEEE, 1986–1990.

[37] Zhuo Wei, Zheng Yan, Yongdong Wu, and Robert Huijie Deng. 2016. Trustworthy authentication on scalable surveillance video with background model support. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 12, 4s (2016), 64.

[38] Thomas Wiegand, Gary J. Sullivan, Gisle Bjontegaard, and Ajay Luthra. 2003. Overview of the H. 264/AVC video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology* 13, 7 (2003), 560–576.

[39] Hua Yang, Jill M. Boyce, and Alan Stein. 2008. Effective flicker removal from periodic intra frames and accurate flicker measurement. In *15th IEEE International Conference on Image Processing, 2008 (ICIP'08)*. IEEE, 2868–2871.

[40] Yun Ye, Song Ci, Aggelos K. Katsaggelos, Yanwei Liu, and Yi Qian. 2013. Wireless video surveillance: A survey. *IEEE Access* 1 (2013), 646–660.

[41] Fan Zhang and David R. Bull. 2011. A parametric framework for video compression using region-based texture models. *IEEE Journal of Selected Topics in Signal Processing* 5, 7 (2011), 1378–1392.
[42] Xiang Zhang, Siwei Ma, Shiqi Wang, Xinfeng Zhang, Huifang Sun, and Wen Gao. 2017. A joint compression scheme of video feature descriptors and visual content. *IEEE Transactions on Image Processing* 26, 2 (2017), 633–647.

Received June 2017; revised February 2018; accepted April 2018