A Two-Stage Intelligent Compression System for Surveillance Videos

KYAW KYAW HTIKE
UCSI University
School of Information Technology
Jalan Menara Gading, Kuala Lumpur
MALAYSIA
ali.kyaw@gmail.com

Abstract: Surveillance videos are becoming immensely popular nowadays due to the increasing usage of surveillance systems in various places around the world. In such applications, cameras capture and record information over long durations of time, which result in large quantities of data, necessitating specialized compression techniques. Conventional video compression algorithms are not sufficient and efficient enough for such videos. In this paper, a novel two-stage compression system for surveillance videos is proposed that can automatically adapt the compression based on the semantic content of the video data. The initial stage consists of an "intelligent interesting event detector" that discards groups of frames in which no interesting events are detected, effectively reducing the size of the video without any loss in video quality. The removal process is robust to minor illumination variations and other small periodic movements. In the second stage, the remaining frames are compressed by HuffYUV codec which is a lossless compression scheme. Results indicate that compression ratios that can be achieved by our system are very encouraging and we demostrate the effectiveness of our system on seven different surveillance videos consisting of a wide range of scenerios.

Key–Words: - Intelligent system, Video processing, Compression, Surveillance system

1 Introduction

Surveillance videos are extensively utilized in domains such as access control in airports [1], traffic monitoring [2] and human activity recognition [3]. In many of these applications, cameras capture information over extended periods of time, ensuing in bulky amounts of data [4, 5]. Such large amounts of records require compression techniques that are not only efficient but also suitable for the domain of surveillance.

Conventional image and video compression techniques are not adequate for compressing for surveillance videos. Compression techniques should take advantage of both spatial and temporal redundancies present in the data. In contrast to spatial redundancy, which has been comprehensively studied in the image compression field, temporal redundancy has received less attention. The majority of video compression techniques are intended for general purpose videos [6, 7, 8, 9, 10]; for example, where no assumptions about scene structure or camera motion can be made. However, for surveillance purposes, videos are typically captured from fixed cameras, resulting in large amounts of temporal redundancy due to high inter-frame relationship. Therefore, proper techniques can be systematically applied to attain very high compression ratios without losing critical and meaningful information.

2 Related Work

Meessen *et al*. [11] propose an object based video coding system using MPEG 2000 in order to store and deliver surveillance video over low bandwidth channels. They attempt to improve the average bitrates or quality ratio of delivered videos when cameras are static. The system developed transmits only Region of Interests (ROIs) of each frame as well as an automatic estimation of the background at a lower framerate in two separate Motion JPEG 2000 streams. This technique allows better video quality and reduced client CPU usage with negligible storage overhead. Unlike [11], our approach does not require splitting into separate motion channels and therefore more efficient and simple to implement, while at the same time achieving much better compression ratios for surveillance videos.

Some authors [12, 13] deal with JPEG 2000 coding or transmission based on the ROI feature and the multi-layer capability provided by this coding system. Those approaches allow delivering more quality for mobile objects (or ROI) than for the background when bandwidth is too narrow for a sufficient video quality. This approach is however different from our work in that it requires setting manually several hyper-parameters that indicate which layer to insert for each ROI feature. This set of hyper-parameters can be sensitive, difficult to set manually and inefficient when there are a lot of uninteresting events that occur in the videos. In contrast, in our work, there are only a few parameters to be set and this can be done in a robust way.

In [14], Liu *et al*. designed a wavelet-based ROI and FOI (Frame of Interest) scheme in order to achieve higher compression ratios. In their design, high priority to ROI or FOI was given by allocating more bits than others. The video compression can be implemented for the cases with some period of clips of low quality and some periods of clips of high quality only at the regions of interest. The advantage of their scheme is that it works not only for arbitrary shaped ROI but also for any combination of different wavelet filters and translation variant and translation invariant wavelet transform. It is also practical to different kinds of filters, with or without down-sampling, and can be combined with many useful techniques. A limitation of their method is that it may be computationally expensive and the resulting compression ratios may not be very high in long surveillance videos.

An object-based video compression system using foreground motion compensation for transmission of surveillance videos was proposed by Babu and Makur [15]. The moving objects are segmented from the background, assumed to be known before hand, by using an edge-based technique. Then, the objects in the current frame are motion compensated according to the previous frame and the resulting error is encoded by a shape adaptive discrete cosine transform. A drawback of such an approach is that it is not robust to incorrect foreground segmentation; therefore, information regarding moving objects might be lost.

Hakeem *et al*. [16] propose another object-based approach, where object models are learned while compression is performed. Each object is modeled by a few principal components obtained by principal component analysis (PCA). As in [15], they assume that the foreground segmentation is given.

Instead of performing compression based on whole objects, Nishi and Fujiyoshi [17] propose a method based on pixel state analysis. Although it is possible to restore the intensity of pixels belonging to moving objects, the location of the whole object is not directly extracted from the compressed data. Additionally, key frames need to be saved every few seconds to adapt to ambient illumination changes. Despite this method taking advantage of the temporal redundancy in both background regions and moving objects by looking at variations over time, the reduction in spatial redundancy is not considered since each pixel is separately encoded.

While [17] considers variations on pixels, the method proposed by Iglesias *et al*. [18] represents an entire frame using its projection on the eigenspace computed from a reference frame. In the case of small variations, only a few coefficients need to be stored. However, when major changes take place in the scene, the eigenspace for large regions needs to be updated. They try to overcome this problem by dividing the video into groups of frames and creating an eigenspace for each, assuming small variations within a group of frames.

Recently, Dey and Kundu [19] propose to extract features from videos coded with high-efficiency video coding (HEVC) to help with foreground extraction and segmentation. Their goal is however different from our work in that they are interested in using a compressed video to help with object segmentation whereas we are interested in video compression as the end goal. Zang *et al*. [20] exploits the static background nature of surveillance videos for the purpose of video compression. However, their approach still takes the uninteresting regions into account in the compression resulting in only about twice the compression ratio compared to standard compression schemes whereas our approach can generate compressed video with significantly better compression ratios.

3 Contributions

Our contribution in this paper is four-fold:

1. We propose a video compression algorithm that can automatically adapt the compression based on the semantic content of the surveillance video data.

2. Our approach is simple to implement and does not require expensive hardware to deploy, potentially enabling it to be implemented on personal computers, laptops or even mobile phones.

3. We formulate and present a novel algorithm for interesting event detection for surveillance videos.

4. The compression algorithm can compress and process videos either online and offline in theory. However, in this paper, we only show empirical results for the offline case. Online can be extended without major changes in the future.

5. Our approach integrates video compression and computer vision techniques to achieve significant compression ratios for surveillance videos. For

certain videos, our algorithm outperforms state-of-the-art methods by several orders of magnitude and for others, it improves the state-of-the-art by at least 6.5 times.

4 Our Method

Let a surveillance video $\mathcal{V}$ be a sequence of T number of frames:

$$\mathcal{V} = \{I_1, I_2, I_3, \ldots, I_T\} \tag{1}$$

where $I_t \in [0,1]^{M \times N}$ is the t-th frame in the video where M and N are the height and width of a video frame respectively. Let $I_t^{(i,j)}$ represents the grayscale intensity value of the row i and column j of the t-th frame. If the frame has color pixels, they are converted to grayscale temporarily. We form a matrix of sets of values for each (i,j) position across all $t = 1$ to $t = T$. That is,

$$\begin{bmatrix} \mathbf{d}^{(1,1)} & \mathbf{d}^{(1,2)} & \mathbf{d}^{(1,3)} & \ldots & \mathbf{d}^{(1,N)} \\ \mathbf{d}^{(2,1)} & \mathbf{d}^{(2,2)} & \mathbf{d}^{(2,3)} & \ldots & \mathbf{d}^{(2,N)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{d}^{(M,1)} & \mathbf{d}^{(M,2)} & \mathbf{d}^{(M,3)} & \ldots & \mathbf{d}^{(M,N)} \end{bmatrix} \tag{2}$$

and the set of values for each matrix entry is given by:

$$\mathbf{d}^{(i,j)} = \{I_t^{(i,j)} \mid t \in \{1, 2, \ldots, T\}\} \tag{3}$$

Therefore, each entry $\mathbf{d}^{(i,j)}$ can be thought of a slice at position (i,j) through the temporal sequence of the entire video $\mathcal{V}$.

We now independently fit a probability distribution on each set $\mathbf{d}^{(i,j)}$. Thus, there will be a total of $M \times N$ probability distributions. Let a prior probability distribution on parameters $\boldsymbol{\theta}^{(i,j)}$ for the probability distribution of $\mathbf{d}^{(i,j)}$ be:

$$p(\boldsymbol{\theta}^{(i,j)}) = \sum_{k=1}^{K} \phi_k^{(i,j)} \mathcal{N}(\boldsymbol{\mu}_k^{(i,j)}, \boldsymbol{\Sigma}_k^{(i,j)}) \tag{4}$$

The prior probability distribution represents the prior belief or knowledge before observing any data. The posterior distribution is given by:

$$p(\boldsymbol{\theta}^{(i,j)} | \mathbf{d}^{(i,j)}) = \sum_{k=1}^{K} \tilde{\phi_k}^{(i,j)} \mathcal{N}(\tilde{\boldsymbol{\mu}}_k^{(i,j)}, \tilde{\boldsymbol{\Sigma}}_k^{(i,j)}) \tag{5}$$

The latent parameters $\tilde{\phi_k}^{(i,j)}$, $\tilde{\boldsymbol{\mu}}_k^{(i,j)}$ and $\tilde{\boldsymbol{\Sigma}}_k^{(i,j)}$ are learnt using the Expectation Maximization algorithm [21] which optimizes the latent variables to obtain the maximum likelihood solution for a statistical model by repeatedly alternating between fixing the latent variables in one set and optimizing another set, and fixing the second set of latent variables and optimizing the first set. Although the resulting solution is only guaranteed to be a local optimum, we run the algorithm several times with different initializations and take the best optimum to maximize the probability of finding an optimum that is close to the global optimum.

After the latent parameters have been obtained, we can write the distribution as

$$p(I_t^{(i,j)} | \boldsymbol{\theta}^{(i,j)}) = \sum_{k=1}^{K} \tilde{\phi_k}^{(i,j)} g(I_t^{(i,j)} | \tilde{\boldsymbol{\mu}}_k^{(i,j)}, \tilde{\boldsymbol{\Sigma}}_k^{(i,j)}) \tag{6}$$

where the function $g(\cdot)$ is defined as:

$$g(I_t^{(i,j)} | \tilde{\boldsymbol{\mu}}_k^{(i,j)}, \tilde{\boldsymbol{\Sigma}}_k^{(i,j)}) = \frac{\exp\left(-\frac{1}{2}\left(I_t^{(i,j)} - \tilde{\boldsymbol{\mu}}_k^{(i,j)}\right)^{\mathrm{T}} \left(\tilde{\boldsymbol{\Sigma}}_{\boldsymbol{k}}^{(\boldsymbol{i},\boldsymbol{j})}\right)^{-1} \left(I_t^{(i,j)} - \tilde{\boldsymbol{\mu}}_k^{(i,j)}\right)\right)}{\sqrt{(2\pi)^k \left|\tilde{\boldsymbol{\Sigma}}_{\boldsymbol{k}}^{(\boldsymbol{i},\boldsymbol{j})}\right|}} \tag{7}$$

The probability density function $p(I_t^{(i,j)} | \boldsymbol{\theta}^{(i,j)})$ derived in Equation 4 can be considered as a variation of the Gaussian Mixture Model which has been used in various machine learning applications [22, 23, 24, 25]. Here, the number of mixture components is equal to K. In order to determine the value K, we adopt the Bayesian Information Criterion (BIC) [26] which not only encourages $p(\boldsymbol{\theta}^{(i,j)} | \mathbf{d}^{(i,j)})$ to fit to $\mathbf{d}^{(i,j)}$ well but also penalizes models of higher complexity.

Now, for each frame $I_t \in \mathcal{V}$, we introduce the *interest criterion* q_t which is computed as follows:

$$q_t = \sum_{i=1}^{M} \sum_{j=1}^{N} p(I_t^{(i,j)} | \boldsymbol{\theta}^{(i,j)}) \tag{8}$$

We then smooth the sequence $[q_t]_{t=1}^{T}$ using exponential smoothing of the form:

$$q_t \leftarrow \alpha \cdot q_t + (1 - \alpha) \cdot q_{t-1} \tag{9}$$

where $0 < \alpha < 1$ corresponds to the *smoothing factor*. This has the effect of reducing any errors or noises in $[q_t]_{t=1}^{T}$. We now form a new video $\mathcal{V}_c$ where

$$\mathcal{V}_c = \{I_t \mid q_t \geq q_{\text{thresh}}\} \tag{10}$$

This corresponds to removing the groups of frames for which the interest criterion q_t is less than q_{thresh}. The threshold q_{thresh} is automatically found in a robust

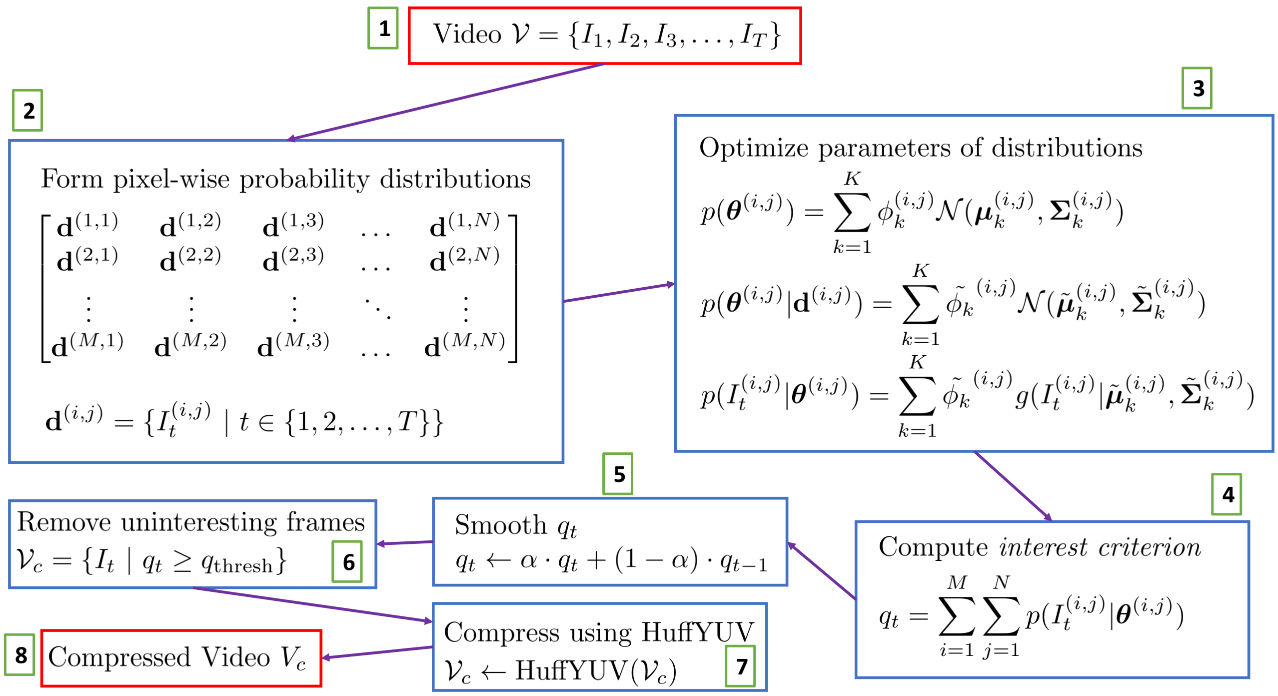

Figure 1: Overview of our proposed intelligent compression system.

	Un	Ours	Huff	Lag
Input_1.avi	896.00	26.30	458.03	349.45
Input_2.avi	69.20	0.03	35.86	26.08
Input_3.avi	527.00	19.50	244.39	222.77
Input_4.avi	65.90	0.02	34.31	26.96
Input_5.avi	728.00	11.90	354.93	283.15
Input_6.avi	491.00	23.80	270.75	197.19
Input_7.avi	547.00	35.40	265.69	230.68

Table 1: Comparison in terms of size (MB) of compressed video output. Un=Uncompressed, Huff=HuffYUV, Lag=Lagarith.

way using Otsu's method [27]. After this, HuffYUV lossless compression is applied on $\mathcal{V}_c$ as follows:

$$\mathcal{V}_c \leftarrow \text{HuffYUV}(\mathcal{V}_c) \tag{11}$$

where the resulting $\mathcal{V}_c$ is a high quality video that only contains meaningful video parts that will be of use for surveillance purposes. The flow chart of our method is shown in Figure 1.

5 Results and Discussion

For this paper, the inputs to our proposed system are surveillance videos. One important requirement for

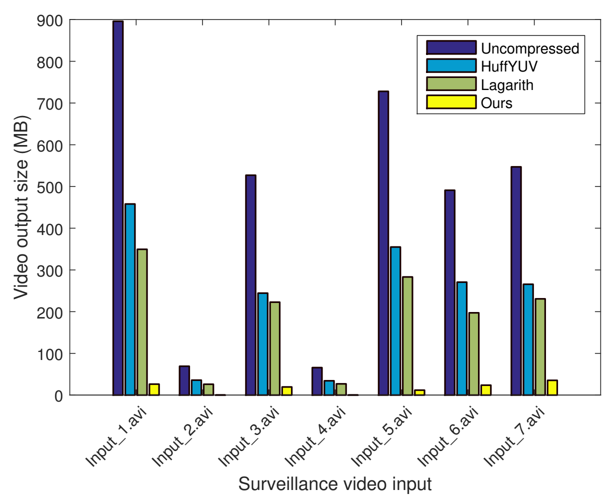

Figure 2: Comparison in terms of size (in MB) of the compressed video output.

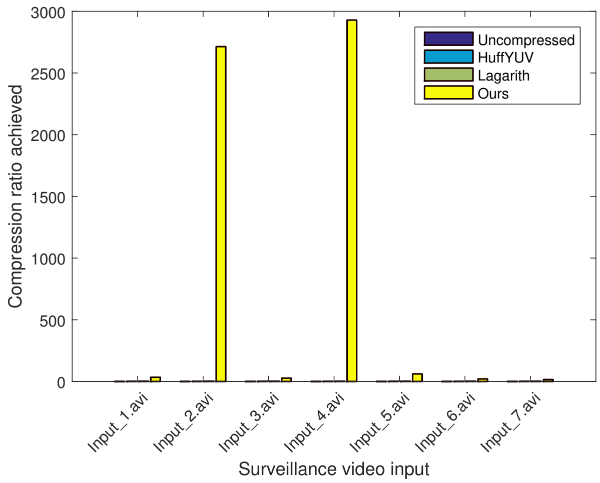

Figure 3: Comparison in terms of compression ratio achieved in the output video.

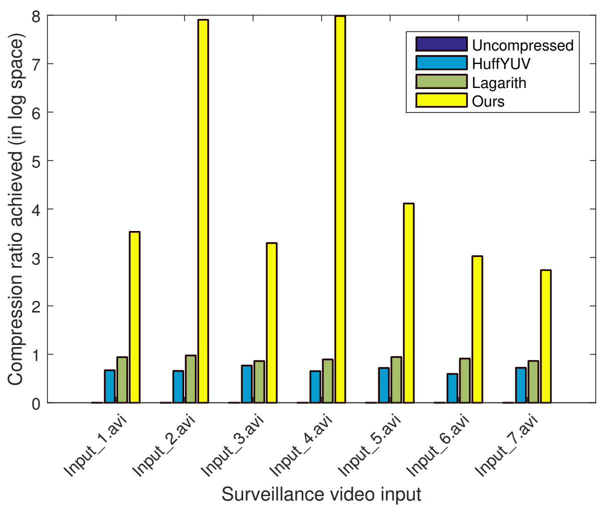

Figure 4: Comparison in terms of compression ratio (in log space) achieved in the output video.

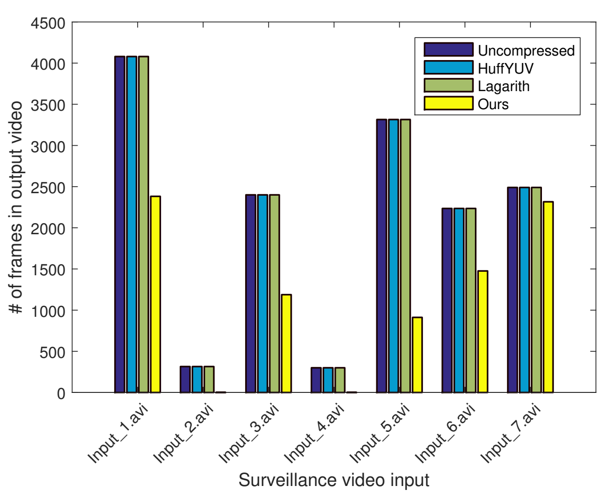

Figure 5: Comparison in terms of the number of frames in the video output.

	Un	Ours	Huff	Lag
Input_1.avi	1.00	34.07	1.96	2.56
Input_2.avi	1.00	2713.73	1.93	2.65
Input_3.avi	1.00	27.03	2.16	2.37
Input_4.avi	1.00	2928.89	1.92	2.44
Input_5.avi	1.00	61.18	2.05	2.57
Input_6.avi	1.00	20.63	1.81	2.49
Input_7.avi	1.00	15.45	2.06	2.37

Table 2: Comparison in terms of compression ratio achieved in the output video. Un=Uncompressed, Huff=HuffYUV, Lag=Lagarith.

	Un	Ours	Huff	Lag
Input_1.avi	4080	2382	4080	4080
Input_2.avi	315	1	315	315
Input_3.avi	2400	1188	2400	2400
Input_4.avi	300	1	300	300
Input_5.avi	3315	912	3315	3315
Input_6.avi	2235	1476	2235	2235
Input_7.avi	2490	2316	2490	2490

Table 3: Comparison in terms of number of frames in output video. Un=Uncompressed, Huff=HuffYUV, Lag=Lagarith.

these videos is that they should be uncompressed in the first place to fully analyze the workings of the compression system. This is the main reason why the input test videos had to be manually recorded and why test videos could not be simply obtained from the Internet or publicly available databases. Although, there are surveillance videos available online, they are not usually suitable as inputs for this research because:

1. They are already compressed to a high degree. Therefore, they are already of low quality and there is no point in compressing any further.
2. If they have not been compressed to a high degree, the file sizes could be extremely large for long videos, making it infeasible to download them from the Internet.

We capture seven *uncompressed* surveillance videos with each video recorded in different scenarios and exhibiting various levels of challenges. All of them have the same container (AVI), frame-rate, video-sample-size and frame resolution. The details of the videos are as follows:

- `Input_1.avi`: Simple scene with humans walking into and out of the scene and conducting various activities such as sitting down.

Figure 6: Frame samples from input survelliance videos.

- `Input_2.avi`: Complex scene with curtain moving with no humans present in front of the camera for the whole video.
- `Input_3.avi`: Complex scene with curtain moving with humans going in and out of the scene.
- `Input_4.avi`: Complex scene with varying illumination and no humans are present in front of the camera for the entire video.
- `Input_5.avi`: Complex scene with varying illuminations and humans going in and out of the scene performing routine activities in an indoor situation.
- `Input_6.avi`: Another complex scene with varying illuminations with humans going in and out of the scene performing house chores.
- `Input_7.avi`: Complex scene with varying illuminations and humans going in and out of the scene, and involving a greater distance between the camera and the scene.

Frame samples from each of these videos are shown in Figure 6. To compare our method with relevant state-of-the-art lossless video compression algorithms, the following experiments are conducted:

- `Uncompressed`: The original videos without any compression.
- `HuffYUV`: The compression algorithm of Ben Rudiak-Gould [28] which is similar to lossless JPEG. In particular, we use HuffYUV 2.1.1.
- `Lagarith`: An lossless open source compression technique by Ben Greenwood [29]. We use Lagarith version 1.0.0.1.
- `Ours`: The lossless intelligent video compression algorithm proposed in this paper (see Section 4).

We use *compression ratio*, β_r, as the main evaluation criterion and it can be defined as:

$$\beta_r = \frac{h(\mathcal{V})}{h(\mathcal{V}_c)} \tag{12}$$

where $h(\cdot)$ is the function to get the size of a video and as defined earlier in Section 4, $\mathcal{V}$ is the input uncompressed surveillance video and $\mathcal{V}_c$ is the resulting compressed video output.

We compare our results (*i.e.* `Ours`) with two most relevant state-of-the-art techniques (*i.e.* `HuffYUV` and `Lagarith`). The baseline is `Uncompressed` which can also be considered as the lower-bound for our study since any compression algorithm should be better than `Uncompressed`, *i.e.* β_r should be at least > 1.

For each of the input surveillance videos, the resulting output video size obtained after different video compression techniques are shown in Figure 2. In order to facilite more precise comparisons, we also give the raw size values in Table 1. From this, it can be seen that for all input surveillance videos `Input_1.avi` to `Input_7.avi`, `Ours` results in the smallest output sizes. In fact, for `Input_2.avi` and `Input_4.avi`, the output sizes are almost zero due to the fact that, as described earlier, these videos do not contain any interesting events (*i.e.* no humans present at all). Therefore, `Ours` has automatically adapted the compression algorithm to exclude these portions in an “intelligent” way.

The compression ratios are compared in Figure 3 and Table 2. From these, it can be seen more clearly that `Ours` significantly outperforms state-of-the-art techniques. In fact, the largest compression ratio obtained by `Ours` is 2928.89 which is much higher than that of the competing approaches (which achieves 2.57). Here, again it can be observed that `Ours` automatically adapts the compression ratio based on the actual contents and semantic meaning of the input videos. Since `Ours` outperforms state-of-the-art by several orders of magnitude, for `Input_2.avi` and `Input_4.avi`, in Figure 3, they dominate the graph. Therefore, we take natural log of the compression ratios and plot these in Figure 4. But these values should not treated as “compression ratios” anymore. However, Figure 4 shows how much effective `Ours` is in compressing surveillance videos.

For completeness, we also show the number of frames in each output video in Figure 5 and Table 3. It can be seen that `Ours` automatically detect the best number of frames to reduce in the output videos achieving great compression ratios (as described earlier) whereas for state-of-the-art compression techniques, they do not have this feature. For `Input_2.avi` and `Input_4.avi`, `Ours` automatically generate only 1 frame for the output videos as there are no interesting events in those videos and only frame is sufficient to summarize the entire videos.

6 Conclusion and Future Work

In this paper, we have proposed an intelligent video compression algorithm that automatically adapts to the content of the input video and that significantly outperforms relevant state-of-the-art techniques. Our method is simple to implement and incorporates techniques from the field of computer vision to the area of surveillance video compression to achieve very encouraging results which have been demonstrated empirically on seven different surveillance videos in a wide range of settings and complexity.

As future work, there are many interesting research directions that can be extended from this paper; firstly, more sophisticated interesting event detections can be investigated to see how much compression ratios are improved. Secondly, there is an opportunity to apply the method proposed in this paper to many different types of surveillance videos such as the ones recorded at airports, shopping malls and parking lots. Moreover, even though our algorithm could potentially be used for outdoor surveillance videos involving various categories of objects and complex activities, there is a chance for a more detailed study on this.

References:

[1] Xiaogang Wang. Intelligent multi-camera video surveillance: A review. *Pattern recognition letters*, 34(1):3–19, 2013.

[2] Bo-Hao Chen and Shih-Chia Huang. An advanced moving object detection algorithm for automatic traffic monitoring in real-world limited bandwidth networks. *Multimedia, IEEE Transactions on*, 16(3):837–847, 2014.

[3] Sarvesh Vishwakarma and Anupam Agrawal. A survey on activity recognition and behavior understanding in video surveillance. *The Visual Computer*, 29(10):983–1009, 2013.

[4] Tiejun Huang. Surveillance video: The biggest big data. *Computing Now*, 7(2), 2014.

[5] Min Chen, Shiwen Mao, and Yunhao Liu. Big data: A survey. *Mobile Networks and Applications*, 19(2):171–209, 2014.

[6] Uma Sadhvi Potluri, Arjuna Madanayake, Renato J Cintra, Fábio M Bayer, Sunera Kulasekera, and Amila Edirisuriya. Improved 8-point approximate dct for image and video compression requiring only 14 additions. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 61(6):1727–1740, 2014.

[7] Guido M Schuster and Aggelos Katsaggelos. *Rate-Distortion based video compression: optimal video frame compression and object boundary encoding*. Springer Science & Business Media, 2013.

[8] Iain E Richardson. *The H. 264 advanced video compression standard*. John Wiley & Sons, 2011.

[9] Guido M Schuster and Aggelos Katsaggelos. *Rate-Distortion based video compression: optimal video frame compression and object boundary encoding*. Springer Science & Business Media, 2013.

[10] Mahsa T Pourazad, Colin Doutre, Maryam Azimi, and Panos Nasiopoulos. Hevc: The new gold standard for video compression: How does hevc compare with h. 264/avc? *Consumer Electronics Magazine, IEEE*, 1(3):36–46, 2012.

[11] Jérôme Meessen, Christophe Parisot, Xavier Desurmont, and Jean-François Delaigle. Scene analysis for reducing motion JPEG 2000 video surveillance delivery bandwidth and complexity. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, volume 1, pages I–577. IEEE, 2005.

[12] Jerome Meessen, C Parisot, C Le Barz, Didier Nicholson, and Jean-Francois Delaigle. Wcam: smart encoding for wireless surveillance. In *Electronic Imaging 2005*, pages 14–26. International Society for Optics and Photonics, 2005.

[13] Victor Sanchez, Anup Basu, and Mrinal K Mandal. Prioritized region of interest coding in JPEG2000. *Circuits and Systems for Video Technology, IEEE Transactions on*, 14(9):1149–1155, 2004.

[14] Chaoqiang Liu, Tao Xia, and Hui Li. Roi and foi algorithms for wavelet-based video compression. In *Advances in Multimedia Information Processing-PCM 2004*, pages 241–248. Springer, 2004.

[15] R Venkatesh Babu and Anamitra Makur. Object-based surveillance video compression using foreground motion compensation. In *Control, Automation, Robotics and Vision, 2006. ICARCV'06. 9th International Conference on*, pages 1–6. IEEE, 2006.

[16] Asaad Hakeem, Khurram Shafique, and Mubarak Shah. An object-based video coding framework for video sequences obtained from static cameras.

In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 608–617. ACM, 2005.

[17] Takayuki Nishi and Hironobu Fujiyoshi. Object-based video coding using pixel state analysis. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 3, pages 306–309. IEEE, 2004.

[18] Héctor J Pérez-Iglesias, Adriana Dapena, and Luis Castedo. A novel video coding scheme based on principal component analysis. In *Machine Learning for Signal Processing, 2005 IEEE Workshop on*, pages 361–366. IEEE, 2005.

[19] Bhaskar Dey and Malay K Kundu. Efficient foreground extraction from hevc compressed video for application to real-time analysis of surveillance bigdata. *Image Processing, IEEE Transactions on*, 24(11):3574–3585, 2015.

[20] Xianguo Zhang, Tiejun Huang, Yonghong Tian, and Wen Gao. Background-modeling-based adaptive prediction for surveillance video coding. *Image Processing, IEEE Transactions on*, 23(2):769–784, 2014.

[21] Tood K Moon. The expectation-maximization algorithm. *Signal processing magazine, IEEE*, 13(6):47–60, 1996.

[22] Zoran Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 2, pages 28–31. IEEE, 2004.

[23] Douglas A Reynolds, Thomas F Quatieri, and Robert B Dunn. Speaker verification using adapted gaussian mixture models. *Digital signal processing*, 10(1):19–41, 2000.

[24] Ming-Hsuan Yang and Narendra Ahuja. Gaussian mixture model for human skin color and its applications in image and video databases. In *Electronic Imaging'99*, pages 458–466. International Society for Optics and Photonics, 1998.

[25] Pedro A Torres-Carrasquillo, Douglas A Reynolds, and JR Deller Jr. Language identification using gaussian mixture model tokenization. In *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, volume 1, pages I–757. IEEE, 2002.

[26] David L Weakliem. A critique of the bayesian information criterion for model selection. *Sociological Methods & Research*, 27(3):359–397, 1999.

[27] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *Automatica*, 11(285-296):23–27, 1975.

[28] Ben Rudiak-Gould. Huffyuv v2. 1.1 manual, 2004.

[29] B Greenwood. Lagarith lossless video codec, 2004.