Movement Detection Using Keyframes in Video Surveillance System

Kyutae Kim, Qiong Jia, Tianyu Dong and Euee S. Jang
Hanyang University
kkt1207@hanyang.ac.kr, jaspery654@gmail.com, prof.euee.s.jang@gmail.com

요 약

In this paper, we propose a conceptual framework that identifies video frames in motion containing the movement of people and vehicles in traffic videos. The automatic selection of video frames in motion is an important topic in security and surveillance video because the number of videos to be monitored simultaneously is simply too large due to limited human resources. The conventional method to identify the areas in motion is to compute the differences over consecutive video frames, which has been costly because of its high computational complexity. In this paper, we reduced the overall complexity by examining only the keyframes (or I-frames). The basic assumption is that the time period between I-frames is rather shorter (e.g., 1/10 ~ 3 secs) than the usual length of objects in motion in video (i.e., pedestrian walking, automobile passing, etc.). The proposed method estimates the possibility of videos containing motion between I-frames by evaluating the difference of consecutive I-frames with the long-time statistics of the previously decoded I-frames of the same video. The experimental results showed that the proposed method showed more than 80% accuracy in short surveillance videos obtained from different locations while keeping the computational complexity as low as 20 % compared to the HM decoder.

1. Introduction

Over the past 20 years, as life has become more complex socio-economically, the need for personal safety and security has soared around the world. Most of the real-time surveillance systems in the 20th century consisted of video streams in which data was transmitted uncompressed using analog sensors. Surveillance systems were difficult to transmit information in various ways at the same time, such as sound and location, and it was costly to construct the entire system. In addition, there were many difficulties in applying complex knowledge extraction algorithms for images such as facial recognition and behavioral analysis. However, advances in technology have led to a clearer acquisition of a wider range of surveillance video. With the emergence of computer algorithms that replace human resources, information that can be obtained from surveillance videos has diversified, requiring a lot of human resources for continuous monitoring.

Surveillance systems have evolved technically over three generations[1]. The first generation was simply for monitoring with analog CCTV systems, and the second generation became more efficient in terms of cost as compression and distribution were developed more efficiently with digital imaging monitoring systems. In the second generation, algorithms that provide semi-automatic functions such as tracking computer vision objects and event warnings while acquiring high-definition images through digital sensors have been introduced into the surveillance system[2]. From the beginning of the 21st century, a fully automated wide-area monitoring system has been pursued with the aim of providing reasoning frameworks and behavioral analysis functions with a third-generation system and integrating multiple sensor platforms at the same time[3].

Surveillance systems are designed to detect changes in moving scenes, the more advanced the system requires higher quality video and more cameras. In order to analyze video systems that track a wide area, it is necessary to distinguish and detect changes in visual scenes with a small number of people and cost. This requires an abstract way to summarize only important information. Video abstraction, which distinguishes only static background images from continuously inputted image frames, is designed to separate target frames and other frames by extracting only keyframes and summarizing videos.

In this paper, we propose a way to classify key information through scene changes and collect summarized information. The proposed method is applied based on the I-frame, which can be called a keyframe, using bitstream data compressed with H.265/HEVC codec. In order to classify specific information in the video, that is, sections with movement, a motion detection method is used to compare the differences between I-frames based on the average mse value. The proposed motion detection method applies the sliding window section defined by a specific window size to the entire bitstream to determine the motion based on the mean square error(MSE) value comparing the two I-frames.

The composition of this paper is as follows. First, Section 2 introduces the existing method of detecting motion in a compressed bitstream. The proposed method is described in detail in Section 3. In Section 4, we provided the experimental results of applying the proposed method. Finally, we conclude this paper.

2. Background

The scene where analysis is required in the video data is divided into sections with and without movement. However, it takes a tremendous amount of time and effort to classify the movement of data in existing systems that requires cost and people just by recording and managing. Therefore, studies have been conducted to detect meaningful information of such large-scale video data faster and less costly. Among them, video abstraction techniques include a still-image abstraction method that extracts a set of static key frames from an video, and a moving-image abstraction method through video skimming that extracts a moving sequence.[4] The still-image method can effectively save the video search time even if some original information is lost by summarizing the set of frames defined as representative frames into a short video in the video decoding process.[5] The moving-image method extracts only the segments of the moving parts of the video and extracts a consistent image summarized for the purpose.[6] In this paper, when analyzing video through still-image abstraction, we use a method to extract and search only key frames from the entire bitstream. In the past, methods such as using the pixel difference value between frames and detecting motion by removing the background [7] were mainly used after decoding all the bitstream sections to be analyzed and then applying modeling. However, with existing technology, it takes time to analyze it and restore it. To solve this, a bitstream-level analysis method was studied. Since video data compressed by a video codec is managed in a bitstream format for transmission and storage purposes, a method for detecting motion in a compressed bitstream without a process of restoration has been studied.[8] However, the method of detecting motion in the compressed bitstream had a problem that the accuracy did not exceed about 50%, and there was a limitation in that it was tested in a limited environment because it was tested on video measured indoors. To calculate the probability of having a bit value per byte among the methods of analyzing bitstream levels [9] did not show a clear difference between the interval with and without motion for real-time large video.

3. Proposed method

3.1. Motivation

The proposed method is to estimate the confidence interval to clearly analyze the movement of real-time videos regardless of day or night. In order to apply it in real time, since it is impractical to gather all the necessary information in real time, we chose to form a reliable sample group to estimate the variance of the population. Based on the fact that the consecutive images in a video sequence are highly correlated, we can assume that the videos used in the experiment have a normal distribution according to the central limit theorem if the sample size is large enough.

In each sample video, the exception is classified by estimating the threshold beyond the confidence interval. Since the threshold of a motionless section may vary depending on the environment, the threshold is estimated using data from the motionless section of day and night for universal application. When comparing data, we calculated the p-value for each mse value to calculate the threshold more accurately because the environment changes have uneven mse values. From sufficient sample data through p-value, we estimated the threshold that can appear in the entire video.

A process as shown in Figure 2 is performed to classify whether each frame contains moving objects based on a specific threshold. First, the MSE difference between two I-frames in the bitstream order is calculated using Equation 1.

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j)-K(i,j)]^2 \tag{1}$$

Equation 2 calculates Z-score as an MSE value between frames.

$$Z = \frac{X-\bar{x}}{\sigma} \tag{2}$$

In this case, subtracting the MSE value X of the current frame and the average MSE value of the entire frame is divided by the standard deviation. In Equation 3, the p-value is calculated using Z-score. Equation 3 is calculated using a cumulative density function (cdf) that integrates a standard normal distribution with an average of zero and a standard deviation of one.

$$P-value = 1-erf(\frac{|z|}{\sqrt{2}}) \tag{3}$$

If the P-value, the probability of coming out of the standard normal distribution to the reference value using the cumulative distribution function, is calculated to be a certain threshold value or less, the frame is determined as outlier and excluded from the reference value calculation process. In this case, since the mse value does not become negative, it is calculated as a one-sided value.

$$FeatureValue = \bar{x} - 2\sigma \tag{4}$$

3.2. Method

The proposed method consists of two steps. First, the proposed method separates only the I-frames from the entire bitstream. This can reduce both the decoding time and the analysis time. Second, the proposed method compares the MSE value as a difference value between the consecutive I-frames to be used as a criterion to determine any moving object between frames. Figure 1 is a graph of some of the mse values of I-frames in the entire bitstream. The orange section of the graph shows frames with motion, and the rest shows frames without motion. Figure 1 shows that the MSE value of the I-frame is relatively low in areas without movement, but the MSE value increases rapidly in areas with movement. Through this, it is possible to estimate the section with movement and the section without movement based on the section in which the MSE value of the I-frame rapidly increases or decreases in the real-time video.

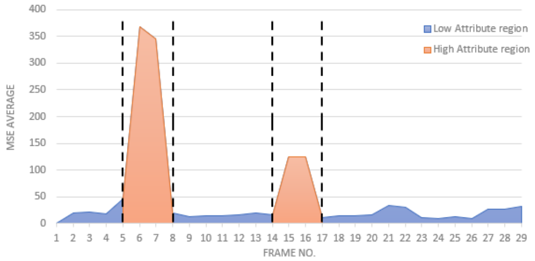
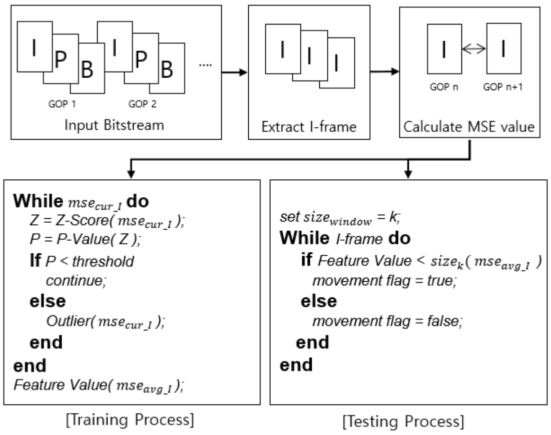


Figure 1. Examples of mse values according to movement

Figure 2 is a diagram of the proposed method. Decodes only the I-frame in the bitstream compressed with the HEVC/H.265 codec in the training step. In the decoded I-frame sequence, the MSE threshold to be used for testing is selected using the difference between the MSE values of the part with and without movement in the surveillance video.

The proposed method analyzes only the I-frame without decoding the entire bitstream for the video surveillance system to classify if there is movement in the video. Based on the feature value calculated through the training process, it is possible to classify exception situations (moving scenes). This allows for less time to determine movement than decoding the entire bitstream.



```
While $mse_{cur\_I}$ do
    Z = Z-Score( $mse_{cur\_I}$ );
    P = P-Value( Z );
    If P < threshold
        continue;
    else
        Outlier( $mse_{cur\_I}$ );
    end
end
Feature Value( $mse_{avg\_I}$ );
```

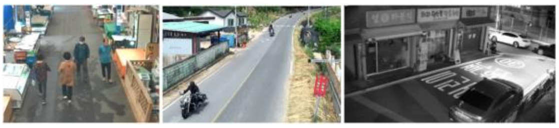
```
set $size_{window}$ = k;
While I-frame do
    if Feature Value < $size_k$( $mse_{avg\_I}$ )
        movement flag = true;
    else
        movement flag = false;
    end
end
```

Figure 2. diagram of proposed method.

4. Experimental Results

We evaluated the proposed method by estimating the accuracy of detecting moving intervals of the proposed method. The test data used three data from different environments encoded in the H.265/HEVC codec. The development environment is Windows 10 with Visual Studio 2019. The hardware is a 3.70 GHz CPU that supports the NVIDIA GTX 750 Ti's 32GB DDR4-2133 MHz memory.

The test video data used in the experiment contains various security camera scenes with and without human movements during daytime or night-time periods. figure 3 is a test sequence. Vehicle sequence is a video of the vehicle's movement on the road during daytime. Person sequence is a video containing the movement of a person during the day. Person-Car sequence is a video of a person's movements at night-time. Table 1 shows information about the video used in the experiment.



(a) (b) (c)

Figure 3. Test Sequence. (a)-Person, (b)-Vehicle, (c)-Person-Car.

Table 1. Test video sequence information.

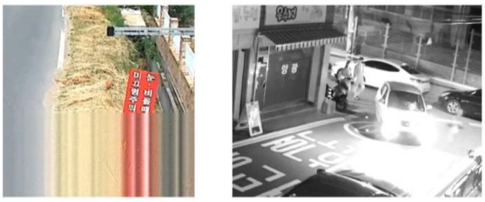
	Person	Vehicle	Person-Car
Frame size	9501	7239	7230
Fps	30	30	30
Data size (KB)	22,105	32,439	37,003
remark	day	day	night

Table 2 shows the accuracy of the experiment with the proposed method compared with the moving interval. In the case of a moving section, the daytime video showed higher accuracy than the night-time video. The motionless sections recorded high accuracy throughout the day and night videos. However, in the case of the vehicle video, the case where it is determined that there is movement in the section without movement was recorded relatively high.

Table 2. Ratio of moving to non-moving intervals.

	Person	Vehicle	Person-Car
Frame with movement	94%	97%	88%
Type 2 error	6%	3%	12%
Frame without movement	99%	80%	98%
Type 1 error	1%	20%	2%

In the vehicle video, there was a case of noise in the video quality as shown in Figure 4-(a), and it was shown that accuracy may be degraded in such an environment. In the case of night videos, some noise may occur as shown in Figure 4-(b), so there seems to be a difference in accuracy.



(a) (b)

Figure 4. Part of the video with noise, (a) Vehicle, (b) Person-Car Sequence.

As shown in Figure 5, the difference between MSE values is large because the noise is greater than that of the daytime video as shown in Figure 6. This can be seen as a result of noise influenced by video quality and surrounding environment, as shown in (a) and (b) in Figures 4, and the accuracy can be improved by correcting the front and rear ranges when determining motion.

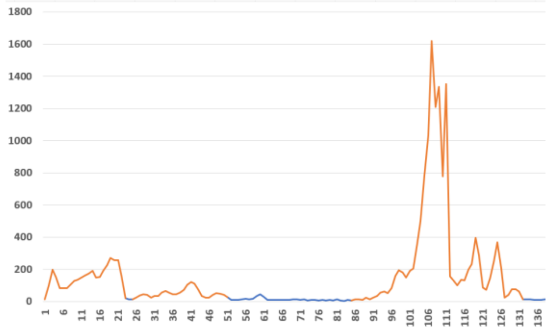


Figure 5. MSE Histogram of Person-Car video

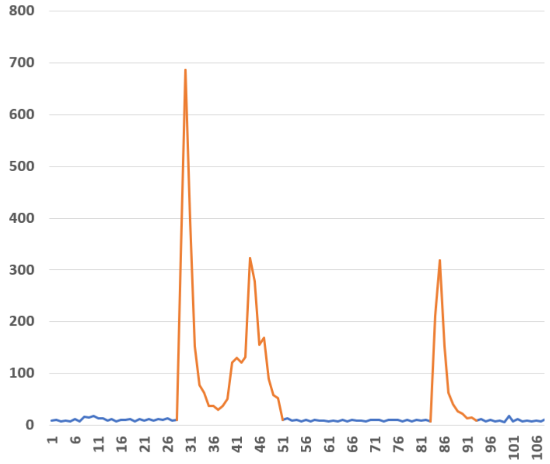


Figure 6. MSE Histogram of Vehicle video

Table 3 shows the computational complexity of the proposed technique. This is the result of decoding only I-frame data using the proposed method compared to the computational complexity of decoding the entire video bitstream. Based on the HM reference software, the calculation complexity of the videos used in the experiment was recorded on average about xx%. Through this, we show the results of reducing the computational complexity of a significant portion. The proposed method does not require a complex algorithm for videos, so it is possible to distinguish motion intervals of videos with such low computational complexity.

Table 3. Computational Complexity Comparison

	Person	Vehicle	Person-Car
reference time	1384.60	967.05	1908.17
tested time	218.32	195.58	243.69
ratio to sec	16%	20%	13%

Previously, techniques for tracking the movement of objects in surveillance videos have been studied intensively. There are many studies that show high accuracy similar to the proposed method in tracking human movement through videos in a limited environment.[10] However, there are insufficient studies that quantitatively evaluate computational complexity suitable for real-time processing, and in most cases only the accuracy of tracking the movement of objects. In this paper, we aim to analyze videos for the operation of real-time large-capacity monitoring systems rather than simple analysis of surveillance videos, so we have a different orientation from methods that require many hardware resources and analysis techniques.

5. Conclusion

The motion detection method proposed in this paper is a method to achieve the goal of existing studies and shows improved results through experimental results. However, further research will be needed on areas with more movement, such as multi-lane roads, vehicles, and crowded places. Further research will also be needed considering changes in accuracy according to the overall video quality and changes in video quality due to various tools of the vehicle and human at night. In the future, various environmental variables will need to be considered for accurate movement detection of real-time video such as CCTV.

6. References

[1] Vassilios Tsakanikas, Tasos Dagiuklas, “Video surveillance systems-current status and future trends”, Computers & Electrical Engineering, Volume 70, pp 736-753, 2018.

[2] D. Cumming, S. Johan, “Cameras tracking shoppers: the economics of retail video surveillance”, Eurasian Business Review, 5 (2), pp. 235-257, 2015.

[3] F. Porikli, F. Brémond, S.L. Dockstader, J. Ferryman, A. Hoogs, B.C. Lovell, S. Pankanti, B. Rinner, P. Tu, P.L. Venetianer, “Video surveillance: past, present, and now the future dsp forum”, IEEE Signal Process. Mag., 30 (3), pp. 190-198, 2013

[4] Truong, B. T. and Venkatesh, S., “Video abstraction: A systematic review and classification.”, ACM Trans. Multimedia Comput. Commun, vol. 3, no. 1, 2007.

[5] Y. Zhuang, Y. Rui, T. S. Huang, S. Mehrotra. “Adaptive key frame extraction using unsupervised clustering”. In Proc. ICIP, 1998.

[6]M. A. Smith and T. Kanade. “Video skimming and characterization through the combination of image and language understanding techniques”. In Proc. CVPR, 1997.

[7] J. M. Kamal Sehairi, Fatima Chouireb, “Comparative study of motion detection methods for video surveillance systems,” Journal of Electronic Imaging, vol. 26, pp. 26 – 26 – 29, 2017.

[8] Yousun Park, Sang-hyo Park, Euee Seon Jang, "Bitstream-based Motion Detection Method for Video Surveillance", International Technical Conference on Circuits Systems, Computers and Communications, 2017.

[9] Min-Ku Lee, Yousun Park, Euee Seon Jang, “Improved motion detection method based on compressed bitstreams”, The Institute of Electronics and Information Engineers, 2018

[10] Jianting Guo, Peijia Zheng, Jiwu Huang, “An Efficient Motion Detection and Tracking Scheme for Encrypted Surveillance Videos”, ACM Trans. Multimedia Comput. Commun. Article 61. Nov 2017.