

# Storage Optimization using Adaptive Thresholding Motion Detection

Muhammad Atif Memon

Computer Science Department  
Sukkur IBA-IET  
Khairpur, Pakistan  
muhammadatif.cs@iba-suk.edu.pk

Sajid Khan

Computer Science Department  
Sukkur IBA University  
Sukkur, Pakistan  
sajidkhan@iba-suk.edu.pk

Zahid Hussain Khund

Computer Science Department  
Sukkur IBA University  
Sukkur, Pakistan  
zahid@iba-suk.edu.pk

Faheem Akhtar

Computer Science Department  
Sukkur IBA University  
Sukkur, Pakistan  
fahim.akhtar@iba-suk.edu.pk

Asif Rajput

Computer Science Department  
Sukkur IBA University  
Sukkur, Pakistan  
asifali@iba-suk.edu.pk

**Abstract-**Data storage is always an issue, especially for video data from CCTV cameras that require huge amounts of storage. Moreover, monitoring past events is a laborious task. This paper proposes a motion detection method that requires fewer calculations and reduces the required data storage up to 70%, as it stores only the informative frames, enabling the security personnel to retrieve the required information more quickly. The proposed method utilized a histogram-based adaptive threshold for motion detection, and therefore it can work in variable luminance conditions. The proposed method can be applied to streamed frames of any CCTV camera to efficiently store and retrieve informative frames.

**Keywords-**storage optimization; adaptive threshold; motion detection; video mining

## I. INTRODUCTION

Image and video processing involves the transmission and storage of large data volumes in the form of frames and images. Several researchers [1-3] worked on reducing bandwidth requirements, whereas others [4] worked on optimizing storage utilization. In today's modern world, a huge amount of data is being generated [5], therefore, extracting the required information from large data volumes is a difficult task, while the storage of large data volumes is an issue [6]. Multimedia mining is a subfield of data-mining that addresses the issues of extracting information such as audio, text, links, images, or digital writing from multimedia files. Many researchers [7-12] worked on video mining, motion detection in CCTV cameras, and crime rate reduction. The importance of CCTV cameras and their placement was discussed in [7], concluding that although placing cameras in both random and expert-suggested sites reduced crime rates, the latter had more impact on crime rate reduction.

Many methods have been proposed for motion detection for security purposes. In [8], MATLAB was used for detecting

motion after getting video images, converting to grayscale, using erosion to remove noise, and finally comparing two consecutive images by using absolute difference. A threshold value between 0.5 and 2 was suggested initially, but further experiments indicated that the ideal threshold value was between 1.3 to 1.5 in most cases. As the use of an image's morphology slows down the processing of every frame, this method cannot be used in real-time video processing. In [9], an external motion sensor and a wireless camera were controlled by an android device. The observer was notified by an email with photos when the motion sensor detected any activity. The suggested system incorporated both an alarm based alert system and a password-based camera power control system. This method's drawbacks were the need for an external sensor for motion detection and the numerous emails in case of continuous motion. A genetic algorithm was utilized for motion detection in [12] to reduce surveillance data storage. The use of genetic algorithms for motion detection resulted in increased complexity and slow frame processing. In [11], motion detection was achieved by calculating the difference between a reference and the current frame. If their difference exceeded a threshold, the frame was marked and transmitted to the observer. Two methods were proposed in [4] for motion-based storage optimization. The first method used a sensor to detect and store the frames containing significant motion. This method had two problems: the use of an external sensor and its range. If the sensor is less sensitive, several sensors are required to cover the whole region of interest. However, if the sensor is more sensitive, it could consider motions out of the region of interest. The second method optimized storage by using neighboring frame differences and a fixed threshold. Fixed threshold-based motion detection will not work well on variable luminance conditions [14-15].

This paper proposes a novel method for optimizing motion detection and video storage without using external sensors. The proposed method uses a single color channel for detecting

Corresponding author: Faheem Akhtar

motion instead of the RGB. Moreover, an adaptive threshold, calculated every 10,000 frames using a frame's histogram, was utilized to detect motion based on significant changes in neighboring frames. The proposed method was faster than the previously proposed methods as it used one color channel and the frame's histogram. Furthermore, the proposed method is dynamic in nature due to its adaptive threshold.

II. METHODOLOGY

Figure 1 shows a summary of the proposed method, while Figure 2 shows the proposed algorithm's flowchart. The adaptive threshold was recalculated every 10,000 frames, and it was applied to the red layer difference of each frame.

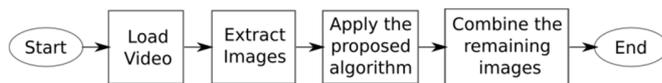


Fig. 1. Summary of the proposed method.

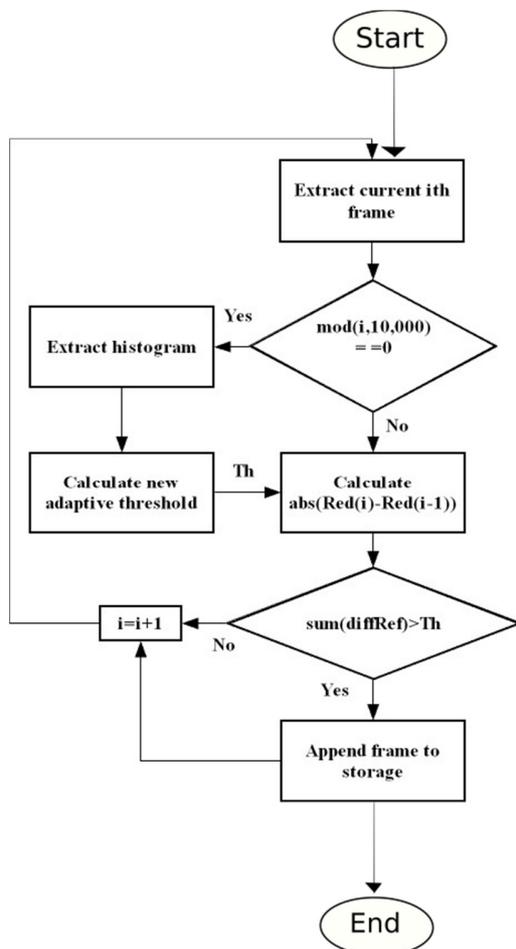


Fig. 2. Flowchart of the proposed algorithm.

A. Using Red Channel for Processing

An RGB frame can be converted to grayscale when the color of the scene is unnecessary. The simpler RGB to grayscale conversion requires two additions and one

assignment per pixel, whereas advanced conversion requires several arithmetic operations [7]. As the proposed method was based on the simple differences of neighboring frames, converting each RGB frame to grayscale would not simplify the algorithm significantly. For example, the conversion of an input frame sized 100×100×3 using exhaustive loops would require 20,000 additions, 30,000 multiplications, and 10,000 assignments per frame. Moreover, the difference calculation requires 10,000 subtractions and 10,000 assignments per frame. So, conversion and difference calculation require in total 30,000 additions/subtractions and 20,000 assignments. On the other hand, using an RGB frame without converting it to grayscale requires 30,000 subtractions and 30,000 assignments. Hence, converting to grayscale would save 10,000 assignments in total for exhaustive differences' calculation.

The proposed method used only the red channel instead of the three RGB or grayscale conversion, as color information was unnecessary. Using the red channel required only 10,000 subtractions and respective assignments for a 100×100×3 sized frame, even if using exhaustive loop-based subtraction instead of vectorization. Moreover, the green channel may be used instead of the red. On the other hand, using the blue channel may not provide good results, as human skin and natural luminance contain fewer blue channel contributions.

B. Adaptive Threshold Calculation

This study aimed to propose a real-time luminance-invariant optimization for motion detection. The luminance of a scene captured by a CCTV camera changes throughout the day due to different atmospheric conditions and daylight changes. If a CCTV camera captures video at 30 frames per second, it takes slightly less than six minutes to capture a ten thousand frame video, while this duration is reduced to three minutes when using 60 frames per second. The first frame after every ten thousand is used for calculating the next threshold. The histogram of this frame is used to calculate the next threshold by using:

$$T_{new} = \frac{\alpha_T}{100} \times B_{80} \times M \times N \quad (1)$$

where  $M$  and  $N$  are the frame's amount of rows and columns, respectively.  $B_{80}$  is the value of  $k$  that can be calculated as the index value that minimizes the difference, given by:

$$B_{80} = \min_{k \in \{20,25,30,\dots,125\}} |N_{80} - \sum_{h=-k}^k F_{P+h}| \quad (2)$$

This corresponds to the frequency of the intensity level in the image's histogram, and it is a count of the 80% of the image's pixels obtained by using:

$$N_{80} = M \times N \times 0.8 \quad (3)$$

$P$  in (2) refers to the index of the peak of the histogram.  $\alpha_T$  is a parameter that controls the sensitivity of motion detection. A smaller value of  $\alpha_T$  will result in more motion detection and vice versa. The possible range of  $\alpha_T$  is:  $0 < \alpha_T \leq 0.17$ . Based on the experience of extensive simulations, the recommended value of  $\alpha_T$  was 0.11. It should be noted that all calculations were made for a non-normalized 8-bit image having intensities between 0-255.

C. Application of Threshold

A frame is marked to have motion if:

$$\Sigma(diffRef) > T_{new} \quad (4)$$

where  $diffRef$  is given by:

$$diffRef = I_R(i) - I_R(i - 1) \vee \quad (5)$$

where  $I_R(i)$  and  $I_R(i - 1)$  are the red channel layers for the current and the previous frame respectively. Each frame having motion is stored in the database, where the others are neglected. Figure 3 shows a test case during the daytime displaying an original frame, a grayscaled frame, and the red channel's histogram. Figure 4 shows a test case during the nighttime, displaying an original frame, a grayscaled frame, and the red channel's histogram.

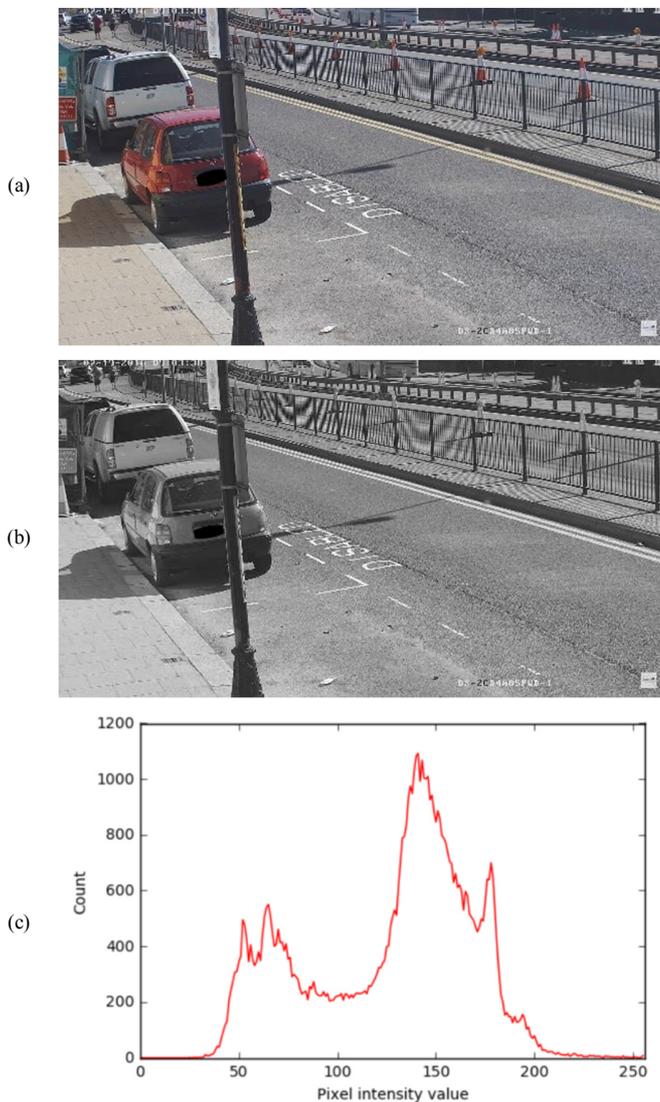


Fig. 3. Daytime case test: (a) Original colored image, (b) grayscaled image, and (c) pixel intensity count of the red channel.

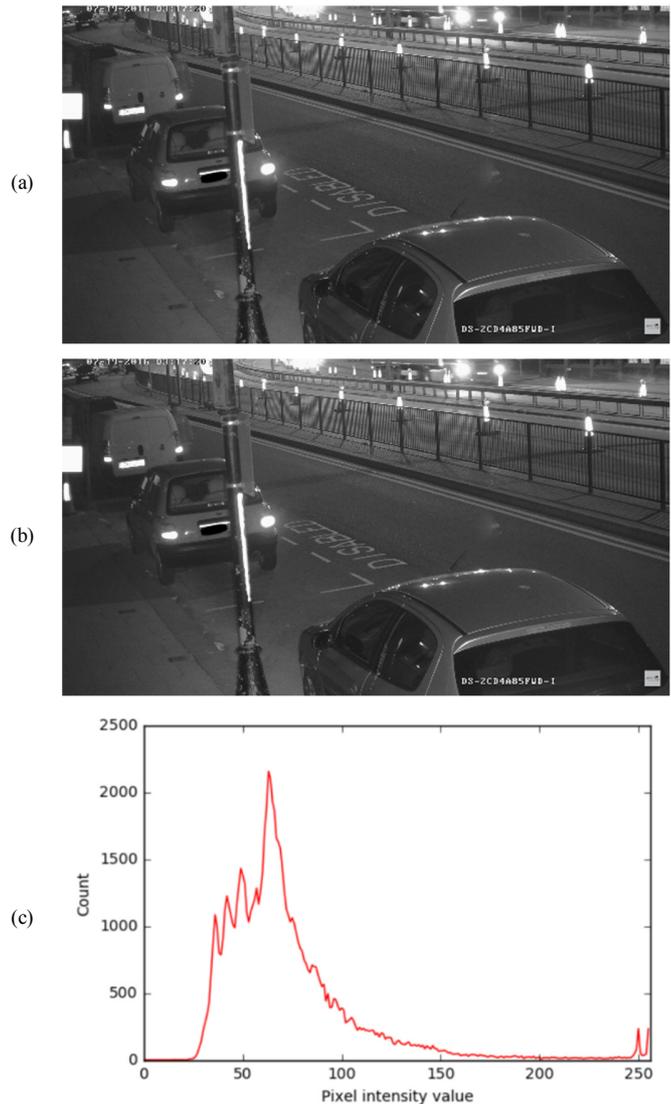


Fig. 4. Nighttime case test: (a) Original colored image, (b) grayscaled image, and (c) pixel intensity count of the red channel.

III. RESULTS AND DISCUSSION

The proposed method was applied on datasets obtained from Sukkur IBA IET (Institute of Emerging Technologies), Khairpur CCTV Cameras, YouTube, and the Venom movie. Results were obtained for both the threshold calculation and storage optimization processes. Figure 3 shows a test case during daytime, used for calculating the threshold. As it can be noted, the histogram was well distributed. Any object moving during the daytime would result in a significant change of intensities and it could be detected using a higher threshold. On the other hand, Figure 4 shows a test case during nighttime. As it can be noted, the histogram is not well-distributed. Any object moving during the nighttime would produce less change of intensities and hence it would require a smaller threshold for motion detection. Moreover, the size of the frame plays a significant role, as a high-resolution frame would require a

higher threshold when compared to a lower-resolution frame of the same scene.

As both 3(a) and 4(a) Figures had 1920×1080 resolution, the total frame's pixels were 2,073,600. The thresholds calculated for the day and nighttime frames were 182,476 and 68,428, respectively. The values of were 62 and 30 for day time and night time frames respectively. When these frames were downscaled to 960×540, the values of thresholds were 45,619 and 17,107 for day and nighttime, respectively.

Table I shows the results of applying the proposed method to the Venom movie and a CCTV camera footage. Venom movie's total duration was 92 minutes, and after applying the proposed method it was reduced to 52 minutes. The total frames of the Venom movie were 11,040, while 6,112 frames were preserved. The frame reduction of this movie was 44.63%.

TABLE I. PERFORMANCE COMPARISON OF THE PROPOSED SCHEME WITH THE ORIGINAL VIDEO

Size	Video sample of venom movie	Video sample of CCTV recording
Actual size	807 MB	14.4 GB
Size after applying the proposed method	220 MB	251 MB

Regarding the application of the proposed method to a CCTV camera stored footage, significant optimization was observed. A video having 268.741 minutes duration was reduced to 32.25 minutes since the CCTV footage had rare motion. The total frames of the input video were 32,249, while after ignoring the redundant frames only 3,714 were preserved. In terms of percentage, the total frame storage was reduced up to 88.4%. The proposed method worked faster than conventional methods by saving many computations due to the use of a single color channel for motion detection. Moreover, the proposed method expedited the calculation process by utilizing frame histograms. Furthermore, the proposed method is dynamic in nature, due to its adaptive threshold and it achieved a significant reduction in storage needs.

#### IV. CONCLUSION

This paper presented a video optimization method that reduced storage needs by ignoring frames without significant information. A significant informative frame is decided based on the proposed motion detection method. The conventional motion detection methods use fixed thresholds that make them sensitive in variable illumination. For efficient processing, the new threshold was calculated using a histogram every 10,000 frames, considering the illumination of the current scene. The absolute difference was calculated between consecutive frames, and its sum was compared with the threshold to decide whether the frame should be stored in memory or not. Furthermore, the red channel of each frame was used, instead of grayscale or RGB in conventional methods. The proposed method provided efficient video optimization, and it can be applied to any streaming CCTV camera irrespective of the radiance of any scene's light source or the illumination of the CCTV camera.

#### REFERENCES

- [1] S. Khan and D. Lee, "Efficient deinterlacing method using simple edge slope tracing," *Optical Engineering*, vol. 54, no. 10, Oct. 2015, Art. no. 103108, <https://doi.org/10.1117/1.OE.54.10.103108>.
- [2] S. Khan, D. Lee, M. A. Khan, A. R. Gilal, and G. Mujtaba, "Efficient Edge-Based Image Interpolation Method Using Neighboring Slope Information," *IEEE Access*, vol. 7, pp. 133539–133548, 2019, <https://doi.org/10.1109/ACCESS.2019.2942004>.
- [3] S. Khan *et al.*, "Image Interpolation via Gradient Correlation-Based Edge Direction Estimation," *Scientific Programming*, vol. 2020, Apr. 2020, Art. no. 5763837, <https://doi.org/10.1155/2020/5763837>.
- [4] S. Arora, K. Bhatia, and V. Amit, "Storage optimization of video surveillance from CCTV camera," in *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)*, Oct. 2016, pp. 710–713, <https://doi.org/10.1109/NGCT.2016.7877503>.
- [5] M. Hazas, J. Morley, O. Bates, and A. Friday, "Are there limits to growth in data traffic? on time use, data generation and speed," in *Proceedings of the Second Workshop on Computing within Limits*, Irvine, CA, USA, Jun. 2016, pp. 1–5, <https://doi.org/10.1145/2926676.2926690>.
- [6] M. H. Padgavankar, "Big Data Storage and Challenges," *International Journal of Computer Science and Information Technologies*, vol. 5, no. 2, pp. 2218–2223, Apr. 2014.
- [7] J. M. Caplan, L. W. Kennedy, and G. Petrossian, "Police-monitored CCTV cameras in Newark, NJ: A quasi-experimental test of crime deterrence," *Journal of Experimental Criminology*, vol. 7, no. 3, pp. 255–274, Sep. 2011, <https://doi.org/10.1007/s11292-011-9125-9>.
- [8] F. N. Tawfeeq, "Real Time Motion Detection in Surveillance Camera Using MATLAB," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, no. 9, pp. 622–626, Sep. 2013.
- [9] M. Rushambwa, T. Chamunorwa, and K. Nyachionjeka, "Real Time Wireless Surveillance System with Motion Detection and Device Control over Internet," *International Journal of Scientific and Research Publications*, vol. 6, no. 3, Mar. 2016, Art. no. 470.
- [10] S. Huang, "An Advanced Motion Detection Algorithm With Video Quality Analysis for Video Surveillance Systems," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 1, pp. 1–14, Jan. 2011, <https://doi.org/10.1109/TCSVT.2010.2087812>.
- [11] A. Upasana, B. Manisha, G. Mohini, and K. Pradnya, "Real Time Security System using Human Motion Detection," *International Journal of Computer Science and Mobile Computing*, vol. 4, no. 11, pp. 245–250, 2015.
- [12] J. C. M. Butil, M. L. F. Magsisi, J. H. Pua, P. K. Se, and R. Sagum, "The Application of Genetic Algorithm in Motion Detection for Data Storage Optimization," *International Journal of Computer and Communication Engineering*, vol. 3, no. 3, pp. 199–202, May 2014.
- [13] C. Saravanan, "Color Image to Grayscale Image Conversion," in *2010 Second International Conference on Computer Engineering and Applications*, Mar. 2010, vol. 2, pp. 196–199, <https://doi.org/10.1109/ICCEA.2010.192>.
- [14] M. V. Sarode and P. R. Deshmukh, "Image Sequence Denoising with Motion Estimation in Color Image Sequences," *Engineering, Technology & Applied Science Research*, vol. 1, no. 6, pp. 139–143, Dec. 2011, <https://doi.org/10.48084/etasr.54>.
- [15] A. B. Altamimi and H. Ullah, "Panic Detection in Crowded Scenes," *Engineering, Technology & Applied Science Research*, vol. 10, no. 2, pp. 5412–5418, Apr. 2020, <https://doi.org/10.48084/etasr.3347>.