2018

VIDEO FRAME REDUCTION IN AUTONOMOUS VEHICLES

Gaurav R. Bagwe
Michigan Technological University, grbagwe@mtu.edu

VIDEO FRAME REDUCTION IN AUTONOMOUS VEHICLES

By

Gaurav R. Bagwe

A REPORT

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

In Electrical Engineering

MICHIGAN TECHNOLOGICAL UNIVERSITY

2018

This report has been approved in partial fulfillment of the requirements for the Degree of MASTER OF SCIENCE in Electrical Engineering.

Department of Electrical and Computer Engineering

Report Advisor:	*Dr. Seyed (Reza) Zekavat*
Committee Member:	*Dr. Michael Roggemann*
Committee Member:	*Dr. Mahdi Shahbakhti*
Department Chair:	*Dr. Daniel R. Fuhrmann*

Table of Contents

List of figures

Acknowledgments

I would like to thank my advisor, Dr. Seyed (Reza) Zekavat for guiding from the very first day of my Masters and helping me to find a direction in my research. I thank him for his constant attention to detail, and for guiding me so patiently even though I made many mistakes. It was a great experience to work with him and he sets an example of hard work.

I thank my committee – Dr. Michael Roggemann and Dr. Mahdi Shahbakhti, who took time from their hectic schedule to evaluate my research and give a necessary update to improve the report.

Further, I cannot thank enough Mojtaba Bahramgiri, a fellow student who helped me in the implementation, design, and testing of the project. Many discussions in the over the past two years helped me polish my work and understand difficult concepts. I am thankful to him for his constant support as a friend.

I would like to thank my friends Vatsal, Kishen, Shreekant, Aditya, Utkarsh, Minorka, Akshara and Himanshu who supported and helped me in the difficult times and became my family at Michigan Tech.

Finally, I thank my parents, and my sister Bhakti, for their faith in me.

Without the contribution from all the people above this report would not have been possible, thank you for their constant support and encouragement.

Abstract

Camera sensors are emerging in many applications such as Smart Buildings and Autonomous driving. The Data generated by multiple cameras in a smart building and autonomous driving applications is usually transmitted through an edge box to a cloud terminal. This transmitted information requires a considerable channel bandwidth, which is not available through current communication standards. The report proposes a Camera Sensor Frame Reduction method to decrease the required channel bandwidth for applications such as autonomous driving.

Here, we propose a method that incorporates cross frame similarity measurement method to reduce the redundant frames and decrease the data rate of each camera. This approach adds processing to the camera sensor, which maps each camera to a smart one. In order to calculate cross frame correlation, each smart camera converts frames into blocks of sub-images. Next, we incorporate consecutive blocks to compute the overall cross frame correlation. The report studies block size selection and its impact on processing complexity and performance. We used real vehicle videos in different driving speed and scenarios to study the complexity and performance of the proposed method. We have investigated frame reduction rate as a function of vehicle traffic and driving environment.

1 Introduction

In recent years, there has been considerable development in the field of autonomous vehicles and Advanced Driver Assistance Systems (ADAS). There are vast benefits of autonomous cars such as improved safety, mobility, reduction in traffic collisions [1]. The vehicles are classified into 0-5 levels of autonomy [2, 3].In level 0, the driver controls all the functions of the car. Level 1, where specific tasks such as either steering or acceleration are done automatically. Level 2, 3 introduces the driver assistance; however, the driver is still always needed to be ready to take over of the vehicle. A vehicle will be able to perform all driving functions in level 4 automation but only under certain conditions. A vehicle of Level 5 will deliver all driving functions in all circumstances. To achieve level 4-5 automation, there will be an increase in the number of sensors to sense the environment, localize the vehicle, and have reliable vehicle-to-vehicle communication[4]. An increase in the number of sensors introduces challenges such as the need for higher processing power and higher channel bandwidth. Specifically, this problem is critical for camera sensors that create a considerable amount of data; This project aims to investigate methods of reducing camera image data.

1.1 Camera Data Volume and Information Content

A camera, 1.3MP and 30 fps, will generate a raw data of 35Mbps. Video Compression techniques such as MPEG-4 or H.264, can reduce the data rate to 8 Mbps. A typical autonomous vehicle level 4-5 in Fig. 1, may use a minimum of 8 cameras that increase the

data rate to 64Mbps[5]. The camera may upgrade to 2MP and 60fps with a combined data rate from all cameras reaching 1Gbps [6]. The current communications standards are unable to handle this information. Consider a vehicle driving through a city in Fig. 2; it contains a significant amount of information such as traffic movement, road signs, road intersections, and most importantly pedestrians. However, a typical highway, shown in Fig. 3 or desert is less informative as it does not have many varying components. When operating the camera at a fixed frame rate, there is some amount of redundant information. This redundant information can be reduced by using a lower frame rate. The frame rate can be a function of speed and information in the environment. The goal of the project is to develop a system to reduce the frames in low informative circumstances.

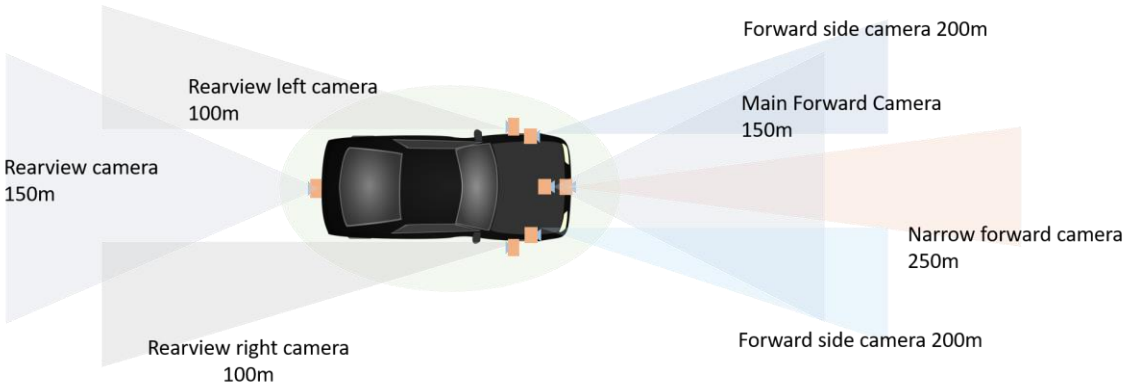

Figure 1 Camera sensors in the Autonomous vehicle.

Figure 2 Urban scenario to drive vehicle.

Figure 3 Highway driving scenario.

1.2 Image Data Reduction Methods

In this section, we discuss the current method available to reduce data generated by the camera sensor. The current work, in the field of object recognition, detects an object and evaluates motion[7]. These motion vectors are used to classify informative and non-informative frames. Another method is to use an inter-frame correlation to detect objects in a video [8]. The objects are used to identify redundant frames. In [9], the authors use Feature Matching to detect motion. These techniques have a high complexity and processing time.

In [10], the author compares template matching and proposes a matching algorithm based on similarity measure using the Sum of Absolute Differences (SAD). The author also compares the similarity measurement techniques such as Sum of Absolute Differences (SAD), Sum of Squared Differences (SSD) and Normalized Cross Correlation (NCC). In [9] the author gives an idea about the methods used in image analysis. The author in [11] compares images using a Joint histogram method. It is a classification technique, which calculates the distance between the images using Histogram. The Drawback of using histogram is that it may produce unstable indication because different images may have the similar histogram [12]. The author in [13] uses Harris-Laplacian feature detector to detect interest point and then compares it with the target Image. The Paper studied the effect of rotation and scaling of the objects based on NCC. The method is useful in matching objects but fails to provide motion of the objects and uses limited features. This method will have

low accuracy on high-resolution images [14]. To address this issue, the image is divided into smaller sub-images.

In [14], the author addresses the issue to process high-resolution images and proposes an adaptive block-processing algorithm. Using Scale Invariant Feature Transform (SIFT) algorithm to match features and identify the objects by dividing the image into smaller images. The author in [15] explains potential improvements in speed by using block partitioning. The method in [15] is used for image compression using parallel processing.

The author [16] uses temporal information in the image to calculate static frames, low motion content in a network for frame dropping. It uses an XOR operation to calculate temporal processing. However, no reference technique has higher chances of false positives, and it tries to address this challenge by use of separate thresholds. The author [17] implements the frame-dropping algorithm, to reduce the slow changing frames, using time-varying hidden Markov model. The author implements this method to reduce the frame in the speech signal using Euclidean distance to calculate similarity.

1.3 The Proposed Technique

We propose a Temporal Block processing method that increases processing speed and reduces the complexity of the existing methods. The frames are preprocessed to mitigate the effects of intensity, illumination, and noise. We divide the frames into smaller blocks of sub-images. We process these blocks independently to find information between recurring frames. We will use Euclidean distance to calculate the similarity between the

blocks. A decision stage compares the number of informative blocks with a threshold to decide to discard or save the frame. The process is repeated through the sequences, and previous discarded frame acts as the reference frame for the next frame.

1.4 The Report Outline

We organize this report in the following style. In Chapter 2, we propose the Cross Frame Similarity Measurement method in detail. We discuss systematic implementation in each section of this Chapter. Chapter 3 is the results of evaluations of the proposed method over test case scenarios. In Chapter 4, we discuss the observations, future scope and conclude the report.

2 The Similarity-Based Data Reduction method

In this section, we discuss the implementation of the proposed similarity-based data reduction method. We divide this Chapter into sections for systematic implementation of the proposed method. In 2.1, we explain the need for block partitioning and the process of its implementation. In the next section, we define a temporal processing method based on the similarity measurement. We also discuss the selection of image distance calculation methods. In section 2.3, we implement a threshold selection based on the Training model. Section 2.4 uses this threshold in the decision stage to detect informative and non-informative image.

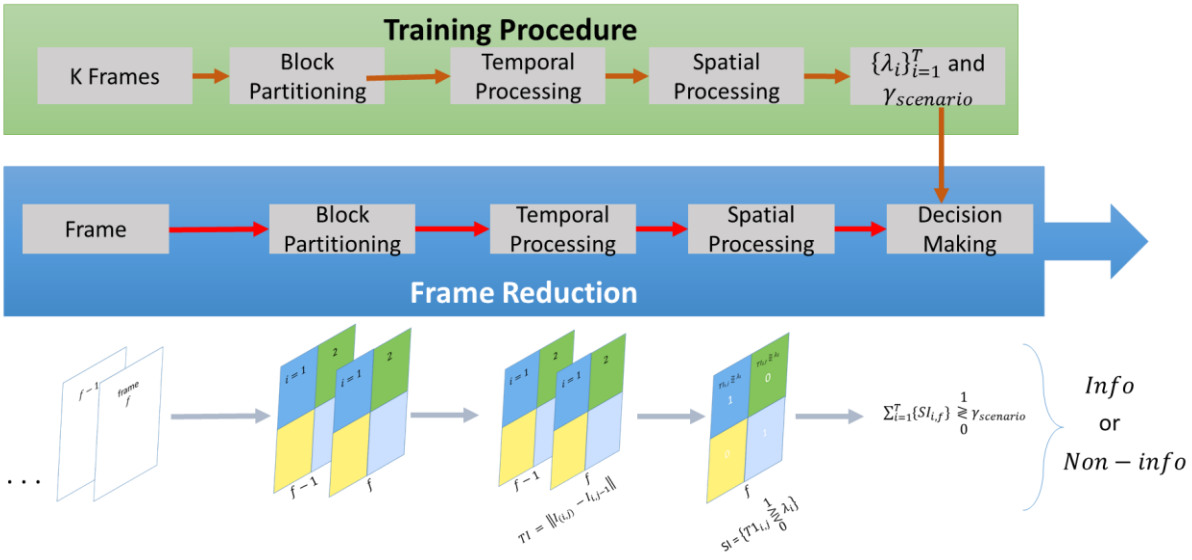

Figure 4 Block diagram of the system.

Fig. 4 represents the block diagram of the proposed method. The video in succession of frames is acquired from the camera sensor. We then pre-process the images to reduce the effect of intensity and noise in the frames. The frames are then divided into sub-images, which are easier to process than the whole image. The temporal processing stage calculates the similarity between the two images and extracts features. These features are then compared with a threshold λ received from the training stage to decide to save or discard the frame. All the blocks are discussed in more detail in the subsequent sections.

2.1 Block Partitioning

Consider an image with the resolution of 1280×1024 (1.3 MP). To measure the similarity of two consecutive image frames, the distance between the two image frames is calculated. The resultant set of features is compared to a threshold based detection method to detect redundant or informative images. Without block partitioning there would be 1,310,720, (1280×1024) set of features making the detection process complex and slow. Block partitioning would reduce the features to T vectors. This reduces the computational time and complexity.

Block partitioning is a method to process the image by dividing it into sub-images to have lower complexity. An image can be divided into smaller images which can be processed independently [15]. It is a widely used technique in satellite image processing, fingerprint matching and video compression [14, 15, 18]. Since we process each block as a separate image, it is possible to compute each block in parallel.

The flowchart of the block partition algorithm is shown in Fig. 5. W, H are the width and height of the image in the frames of the video. Consider we divide the image into blocks represented by $N \times M$, and the total number of blocks is T. We crop each block using the dimension of the rectangle obtained from this partitioning method. The rectangle is 1×4 matrix represented as (r, c, w, h), where, r (row), and c (column) is the location of the first element, and w, h are the width and height of the rectangle. The dimensions of each rectangle are calculated and stored in a matrix of dimension that is $T \times 4$. Each row in the matrix represents the dimension of the block. Subsequently, $1 \times T$ will the dimension of the features and the thresholds in the future stages.

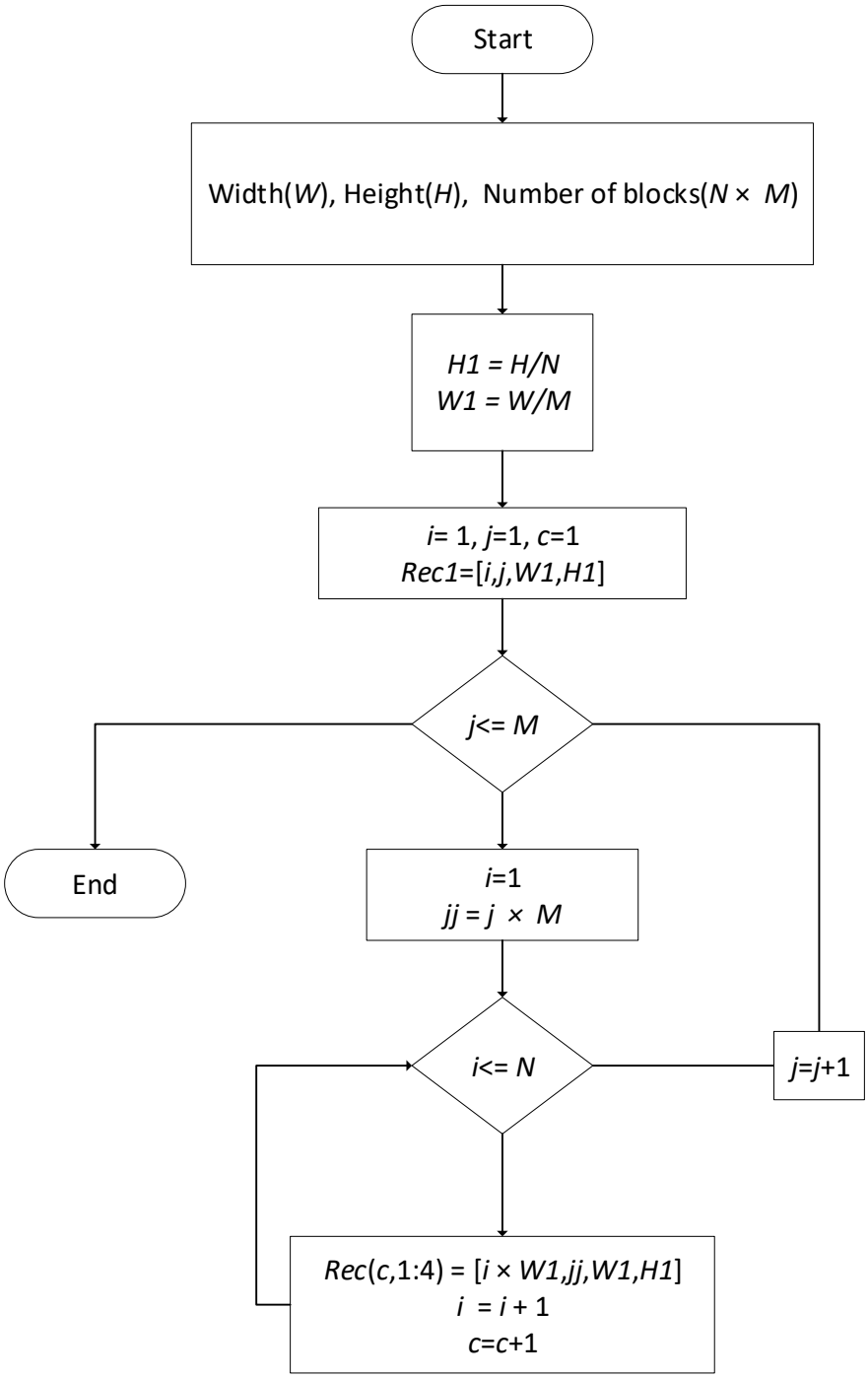

Figure 5 Flowchart of partitioning the blocks in images.

A sub-image is a two-dimensional matrix with width ($W1$) × height ($H1$) pixels as shown in Fig. 6. The row of the Rec represents the block number and columns represents the dimension of the rectangle used to crop the sub-image.

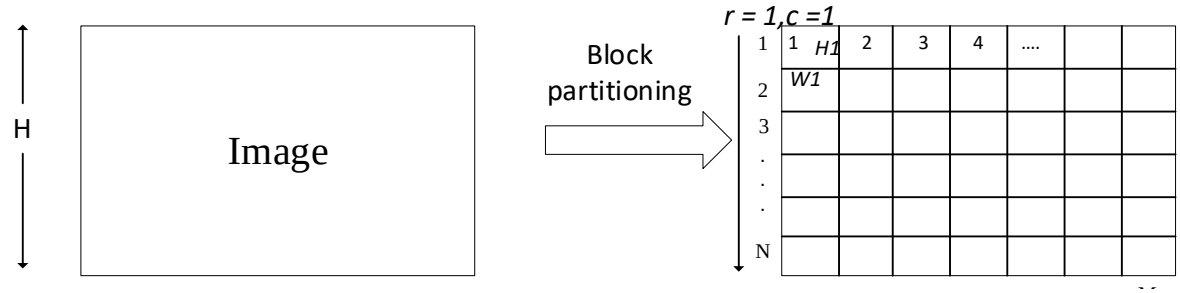

Figure 6 Image and its subsequent blocks.

To process the images in real time, the processing time of the system should be in the order of 0.033 secs for a 30fps video. However, the accuracy of detection of motion is inversely proportional to the size of blocks. Therefore, when we use block partitioning, there is a tradeoff between processing time and the accuracy of the similarity calculation.

2.2 Temporal Processing

As explained in [19], *'There are many ways to estimate the amount of motion content in a video sequence, such as the Difference of Histogram, Block histogram difference, histogram of difference image, and block variance difference. All these measures need extra calculation.*'

The use of histogram may be suitable for clustering or classification of the images [11]. Using Histogram method, we would be able to classify the blocks; however, it would not be able to detect motion in the objects. After considering template-matching techniques, and motion vector based similarity detection, we found temporal processing to be ideal for this application in autonomous vehicles.

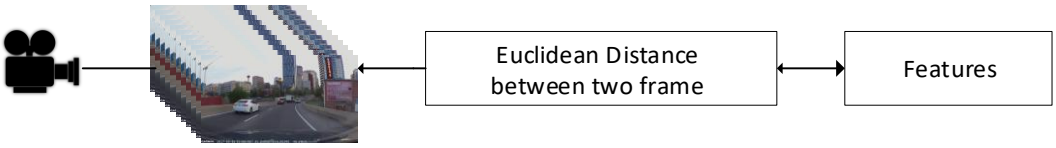

Figure 7 Temporal information between frames using Euclidean Distance.

Consider a video sequence with series of frames $\{f_i\}_{i=1}^{n}$ as shown in Fig. 7. Temporal processing is an analysis of the data varying over time [20]. The features of image in frame f_i are compared with the features of image in frame f_{i-1}. Temporal processing is used to detect the significant motion in a video sequence.

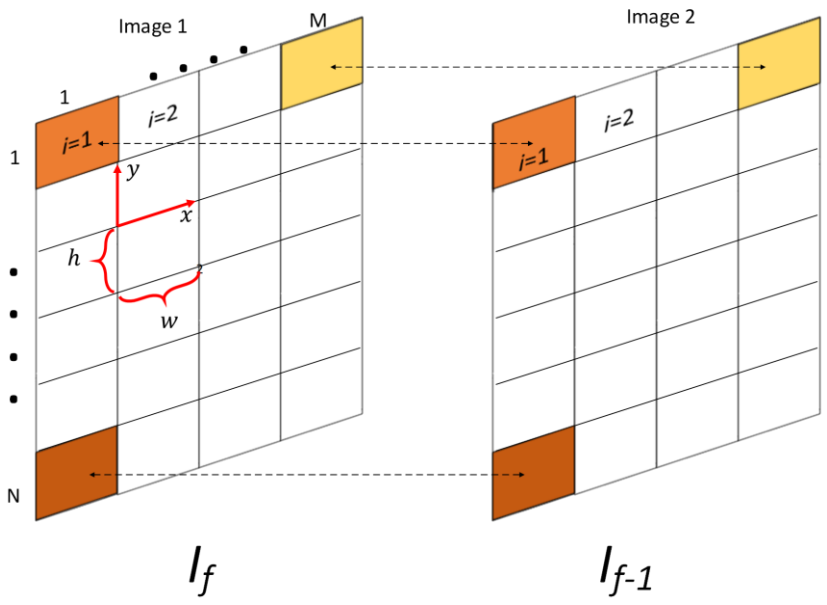

Figure 8 Temporal Information between two frames at i and $i-1$.

Temporal Information (TI) is calculated as the distance between the two consecutive images as shown in Fig. 8. We can calculate the similarity using correlation function. The correlation function has higher time complexity than algorithms like SAD or SSD. Feature detection techniques such as SIFT, SURF, and Features from Accelerated Segment Test (FAST) use SAD and SSD. SAD and SSD are highly reliable, fast and have low complexity. Another method to detect similarity is Normalized Cross Correlation (NCC) which is sensitive to image calculations and but has higher computational complexity. To calculate the similarity of two consecutive frames, f and $f-1$, we compute SSD_f that is defined as the summation of the square of the elementwise difference in the recurring images and corresponds to:

$$SSD_f = \sum_{(x=1,x=1)}^{(W,H)} (I_f(x,y) - I_{f-1}(x,y))^2 \quad (1)$$

Here, $I_f(x,y)$ and $I_{f-1}(x,y)$) are the images in the video at frame f and $f-1$. W and H are the width and height of the image. SSD_f is the similarity measure at frame f compared with frame $f-1$. Similarly, $SAD_f$is the summation of elementwise absolute difference of the two images, and is defined as:

$$SAD_f = \sum_{(x=1,x=1)}^{(W,H)} |I_f(x,y) - I_{f-1}(x,y)| \quad (2)$$

Here, $I_f(x,y)$ and $I_{f-1}(x,y)$ are the images in the video at frames f and $f-1$, respectively. W and H are the dimensions of the image as explained in (1). SAD_f is the similarity measure of frame f when compared with frame $f-1$. The resultant distances are a scalar value and the distance calculation methods will change the threshold linearly. Hence, for simplicity of representation, we use Euclidean distance to calculate distance. Euclidean distance is the square root of SSD. Temporal Information represents the similarity measurement defined by the Euclidean distance between blocks at the same position within consecutive frames. Consider a block i as shown in Fig. 8 at frame f. This block is compared with block i at frame $f-1$ to find the temporal information. Therefore, the temporal information between two images is expressed as:

$$TI_{i,f} = \sqrt{\sum_{(x=1,y=1)}^{(n,m)} (I_{i,f}(x,y) - I_{i,f-1}(x,y))^2} \quad (3)$$

where, $I_{i,f}(x,y)$ and $I_{i,f-1}(x,y)$ are the sub-images cropped by block i at frame f and $f-1$. $TI_{i,f}$ is the temporal information within the block i for the frame f.

We compare this $TI_{i,j}$ with threshold λ_i obtained from the training procedure to decide if the block i is informative. The resultant is the Spatial information which is defined as:

$$SI_{i,f} = \begin{cases} 1 & if\ TI_{i,f} \geq \lambda_i \\ 0 & if\ TI_{i,f} < \lambda_i \end{cases} \quad (4)$$

where $SI_{i,j}$ is the spatial information of block i at frame f, and is 1 if the $TI_{i,j}$ is greater than the threshold λ_i else 0. Using this spatial information we can make a decision to save or discard the frame at the decision stage.

2.3 Training model

The training model is based on Classification problem in machine learning. Each block is classified into redundant and informative based on the TI, defined in (3), between the recurring images. To make a decision, a threshold is computed to find an optimum value to evaluate the occurrence of an event within a block. If TI is more than the threshold, we classify the block as informative and vice versa.

After block partitioning, the image is divided into T blocks. To detect an event in a block i we need to calculate a threshold for the block in the image. To do this, we generate frames in random and calculate the Euclidean distance between the consecutive frames within block i. Consider a block i obtained from the block partitioning stage as shown in Fig. 8.

To incorporate the threshold for detecting an object within the block, K frames are taken into consideration and the cross-frame Euclidean distance is calculated for all K frames. We can calculate the threshold of the image block i, as the average of the Euclidean distances across all K frames that corresponds to:

$$\lambda_i = \frac{\sum_{k=1}^{K} TI_i(k)}{K} \tag{5}$$

where, $TI_i(k)$ is the Euclidean Distance of block i with the same block at its previous frame, and K is the total number of frames. We repeat this process to calculate the threshold for all the blocks. This results in a threshold for each block within the image. Therefore, the dimension of λ is $1 \times T$.

We use $\gamma_{scenario}$ to distinguish between informative or non-informative frames. The $\gamma_{scenario}$ is average of the sum of TI in a particular scenario and is defined by the following equation:

$$\gamma_{scenario} = \frac{1}{\mathrm{K}} \sum_{j=1}^{K} \left(\sum_{i=1}^{T} \{SI_{i,j}\} \right) \tag{6}$$

where, K is the total number of training frames, T is the total number of blocks after block partitioning and $SI_{i,j}$ is the spatial information of block i at frame f.

2.4 Decision Making

The thresholds λ ($1 \times T$) are compared elementwise to the TI obtained from the temporal processing block. If the TI at block i obtained from (3) is greater than threshold λ_i, we

detect an event else the block is non informative. Accordingly, this decision making process can be summarized via:

$$\lambda_i \underset{0}{\overset{1}{\lessgtr}} \sqrt{\left(\sum_{x=1,y=1}^{Y}(I_{i,f}(x,y) - I_{i,f-1}(x,y)\right)^2} \tag{7}$$

where, T is the total number of blocks, $I_{i,f}$ and $I_{i,f-1}$ are Image matrices of frame f at block i. and $\{\lambda_i\}_{i=1}^{T}$ are the threshold for block i. Thus, the output of (5) is Boolean of $1 \times T$ dimensions. Each block which is less than the threshold λ is discarded whereas more than threshold λ in informative and is saved. The number of informative blocks defers depending on the scenario, Highway or City, and traffic speed.

Once we get spatial information from (4) we use the following equation to calculate informative and non-informative frames:

$$Info(f) = \sum_{i=1}^{T}\{SI_{i,f}\} \gtrless \gamma_{scenario} \tag{8}$$

where $Info(f)$ is the information of frame f, which is 1 (informative) if the sum of $SI_{i,j}$ is greater than threshold $\gamma_{scenario}$, else it is 0 (non-informative). The $SI_{i,j}$ is same as defined in (4).

Therefore, we have the informative frames which will be saved and the non-informative frames which will act as the reference frame when comparing the new image.

3 Results

In order to study the performance of the proposed data reduction method, a video [21] from a dataset is taken and divided into training and the testing segments. We implemented the test on MATLAB running on an Intel Xeon 3.70GHz processor with 16Gb ram. The test video consists of 17341 frames out of which we randomly selected 1734 for calculation of the threshold. The testing video includes scenarios of a vehicle in no motion, the vehicle moving in low traffic and high traffic.

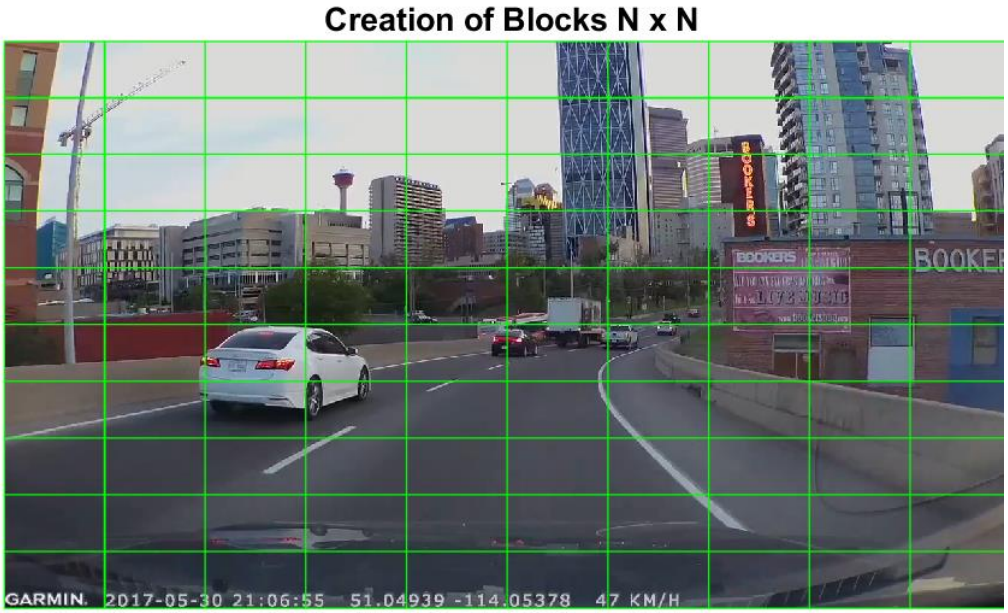

Figure 9 Image divided with Block partitioning.

Fig. 9 is an output of the image after dividing it into sub-images. We divide the image into 100 blocks as shown in Fig. 6. Each block shown in the Fig. 9 is processed as a separate image. The accuracy and the processing time is sensitive to the number of blocks used.

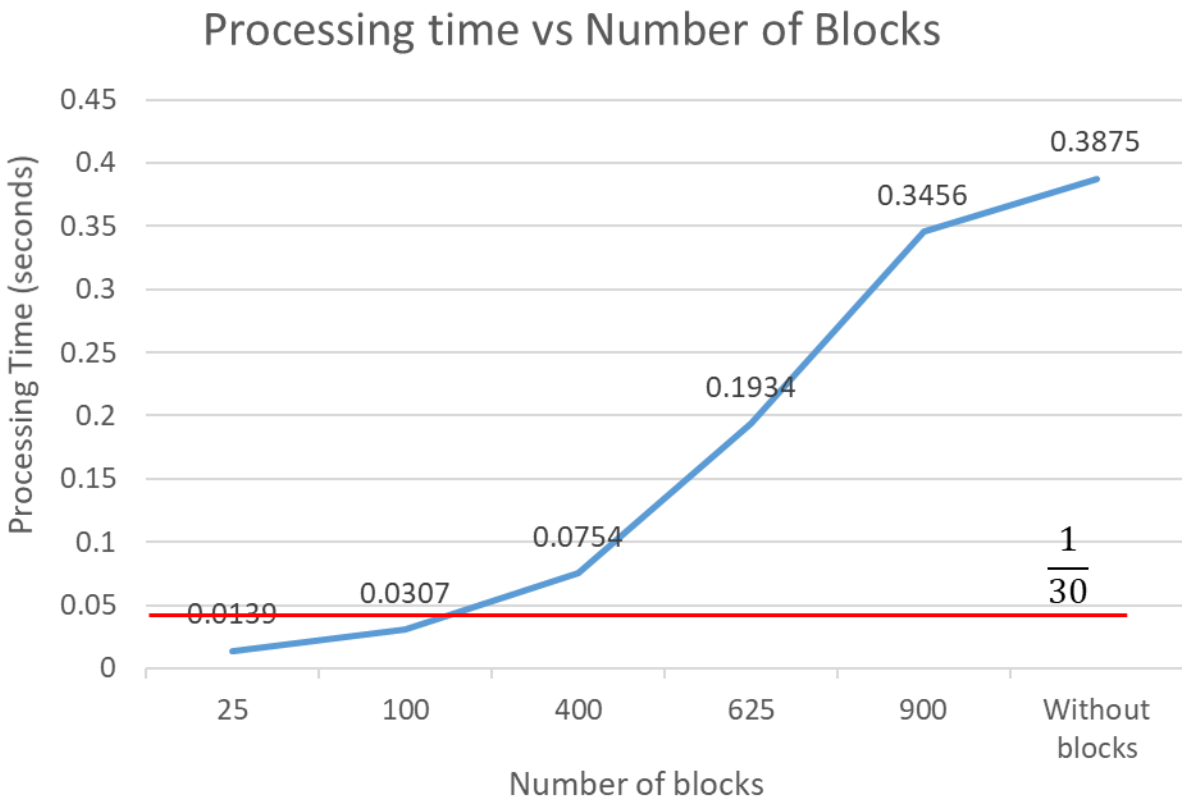

Figure 10 Processing time of one frame vs. number of blocks.

Fig. 10 shows the processing time for (5) vs. the number of blocks. If we process the image without using the blocks, the processing time for 1.3MP camera sensor is 0.3875 secs. We can process the same amount of data 12 times faster by using 100 blocks. Table 1 displays the effect of the using the number of blocks on the accuracy. Partitioning the image into more blocks will result in better accuracy. However, it may result in more processing time making it difficult to implement in real time. There is a trade-off between the accuracy and processing time when using block partitioning. The accuracy itself is less sensitive to the threshold as it can be seen in Table 1. With the use of a greater number of blocks the change

skipped frames is not significant and hence the fewer blocks can be used to give better accuracy.

Using Fig. 10 and Table 1 we can select the optimum number of blocks. In this example, we have considered a video with 30 fps frame rate. Therefore each frame needs to be processed within $\frac{1}{30}$ i.e. $0.033\ secs$. Therefore from Fig. 10, we can use maximum use 130 blocks. The Table 1 we can observe that the difference in accuracy will be small with an expense of greater difference in time complexity. Therefore, we select the image to be partitioned into 100 blocks.

Table 1 Effect of the number of blocks on the accuracy.

Number of blocks	Saved frames in video sequence having redundant information	Saved frames in video sequence having important information
Without Blocks	6.12%	68.74%
25 Blocks	11.85%	89.17%
100 Blocks	13.58%	91.83%
400 Blocks	14.15%	90.23%
900 Blocks	16.06%	89.88%

We selected one of the blocks to display the calculation of the threshold for the particular block. We calculate the Euclidean Distances as a temporal information measure using (3). The Euclidean distance of the block is represented in Fig. 11. The values closer to zero are highly correlated and contain redundant sub-image. The data is highly weighted towards zero. Analysis by synthesis is carried out to find the threshold, and statistical mean divides the blocks into redundant and informative Fig. 12 and Fig. 13.

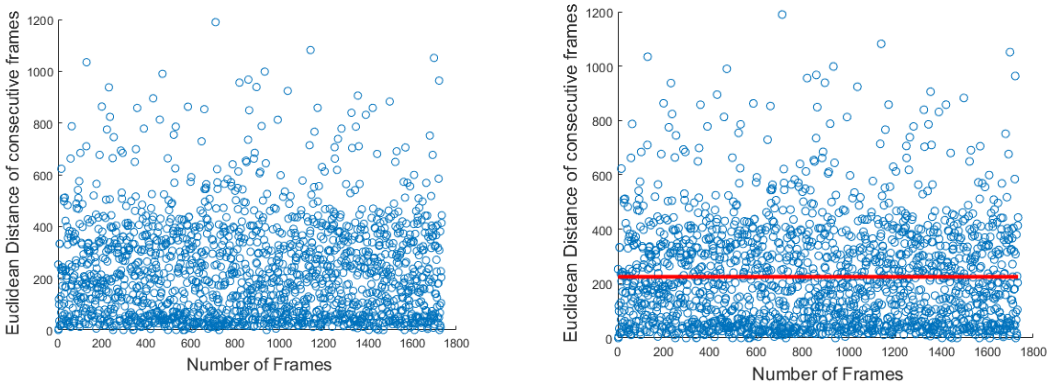

Figure 11 Euclidean Distances of a block over consecutive frames.

Fig. 12 represents the blocks with distance less than the threshold, which are detected as redundant information. The information content in the images of Fig. 12 have less information than the images of Fig. 13. Fig. 13 consists of the images that are higher than the threshold and are informative.

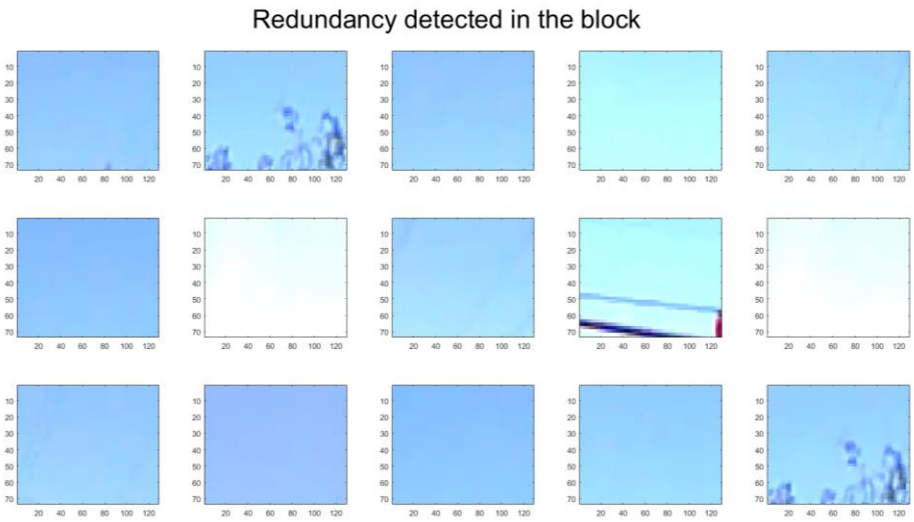

Figure 12 Detection of redundancy in the block.

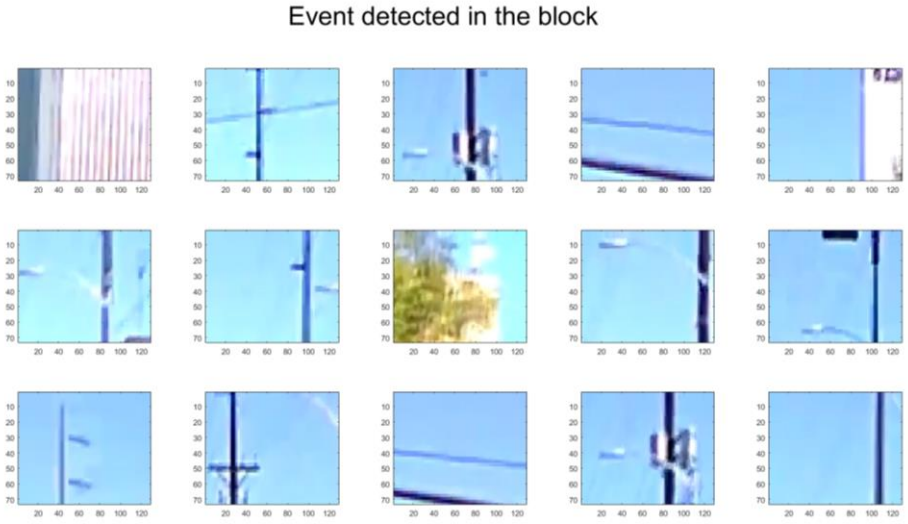

Figure 13 Detection of the events in the block.

After the threshold is calculated for all the blocks using (5) and (6), these thresholds are used to detect redundant frames in video. To test the performance of the system, we create test scenarios in different driving conditions and environments. The test scenarios constitute of a halted vehicle, vehicle in low traffic and vehicle in high traffic.

In test 1-2, we selected a case where the vehicle has no or little motion. The method reduces the redundant frames when the vehicle is stopped and saves the frames only when the vehicle shows movement. The frame rate in such a situation can be set to minimum to save processing power and channel bandwidth.

Figure 14 Video sequence of the output of test 2.

Figure 15 Vehicle driving on a road with low traffic.

Test 3-4 shown in Fig. 15 contains a vehicle moving in light traffic. Such a scenario contains fewer vehicles and would need a lower frame rate. We were able to reduce the frame rate to 20-24fps shown in Table 2.

Figure 16 Vehicle moving in high traffic.

Test 5 shown in Fig. 16 consists of vehicle in high moving traffic, which is more informative and a higher frame rate is required in such a scenario. Since it contains more information than the other test, more frame rate is required to represent the information. We were able to reduce the frame rate to only 28fps.

Table 2 Result of the testing scenarios.

Video	**Description**	**Percentage of skipped frames**	**Optimum frame rate**
Test1	Stopped at Traffic Signal	71.55 %	9 fps
Test2	Vehicle stopped with no traffic	90.63 %	3 fps
Test3	Vehicle moving on a clear road	27.27%	22 fps
Test4	Vehicle moving in light traffic	22.93%	24fps
Test5	Vehicle moving in high traffic	8.96 %	28 fps

We create test scenario to check the algorithm and the results are shown in Table 2. When moving in high traffic we could still reduce the frames by almost 9% and by almost 90% when there is no motion of the vehicle. Test 1 and 2 are similar while Test 3 and 4 are similar with respect to the vehicle speed. Therefore, We can also see from Table 2 that the frame rate should be a function of the vehicle speed as well information change in the environment.

4 Conclusions and Future Work

This report presents a solution to the high communication bandwidth required for autonomous vehicles, as they need to process and send a large number of image data. The report presents a high-performance frame reduction method, which can be used to reduce data in real-time applications such as Autonomous vehicles and connected vehicles.

This solution is achieved by removing redundancy in specific scenarios. The method can reduce the required frame rate from a camera sensor by 30% in scenarios such as a vehicle driving in low traffic or on a highway. It is observed that the processing time taken by the blocks is lower than the processing without using the blocks. We can exploit the block processing method coupled with motion estimation using motion vector to improve the performance of the system.

The method verifies the need for higher frame rate in vehicles moving in high traffic, changing directions, or moving through downtown. It demonstrates that the frame rate can be reduced in certain circumstances. Typically, high frame rate scenarios such as downtown are more populated and have additional access points to accommodate extra bandwidth. They may also use methods such as carrier aggregation to utilize unlicensed band for additional bandwidth. However, low frame rate scenarios are places such as desert, forest or highway, which do not need higher bandwidth and indeed such bandwidth is not available to them. Data reduction method can be used to transmit only the necessary data in such locations.

There are many problems that can be investigated to improve the proposed method. We can estimate the threshold used to compare TI by using methods such as Expectation-Maximizations (EM) to cluster the data. Certain block according to their spatial position may generate false alarm in the sense that the content is declared informative for vehicles, while they are not really informative. These blocks are typically located at the frame edges, which may detect movement of trees, and building as informative. We can also use spatial processing to find the information entropy of a block of sub-image to reduce the erroneous data. We can implement stereo cameras to calculate depth and detect objects, which will improve the system by adding spatial processing. We can further use data from the Cloud Terminal and Global Positioning Systems (GPS), to optimally determine event detection threshold within the processor for a particular area and share the threshold of the scenario with an Autonomous Vehicle.

5 Reference List

1. Hamid, U.Z.A., et al., *Current Collision Mitigation Technologies for Advanced Driver Assistance Systems â€“A Survey.* PERINTIS eJournal, 2016. **6**(2).
2. Gert Rudolph, U.V., *https://www.sensorsmag.com/components/three-sensor-types-drive-autonomous-vehicles.* Nov,2017.
3. Zanchin, B.C., et al. *On the instrumentation and classification of autonomous cars*. in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. 2017.
4. NHSTA, *https://www.nhtsa.gov/technology-innovation/automated-vehicles-safety*.
5. *https://www.networkworld.com/article/3147892/internet/one-autonomous-car-will-use-4000-gb-of-dataday.html*. 2016.
6. Bielby, R., *https://www.micron.com/about/blogs/2017/may/adas-camera-requirements*. 2017.
7. Yokoyama, T., T. Iwasaki, and T. Watanabe. *Motion Vector Based Moving Object Detection and Tracking in the MPEG Compressed Domain*. in *2009 Seventh International Workshop on Content-Based Multimedia Indexing*. 2009.
8. Rout, D.K. and S. Puhan. *Video object detection using inter-frame correlation based background subtraction*. in *2013 IEEE Recent Advances in Intelligent Computational Systems (RAICS)*. 2013.
9. Desai, H.M. and V. Gandhi, *Real-time Moving Object Detection using SURF*.
10. Fouda, Y., *One-dimensional vector based pattern matching.* arXiv preprint arXiv:1409.3024, 2014.
11. Pass, G., *Comparing images using joint histograms.* Multimedia systems, 1999. **7**(3): p. 234.
12. Di Gesù, V. and V. Starovoitov, *Distance-based functions for image comparison*. Vol. 20. 1999. 207-214.
13. Feng, Z., H. Qingming, and G. Wen. *Image Matching by Normalized Cross-Correlation*. in *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*. 2006.
14. Lee, I.H. and T.S. Choi, *Accurate Registration Using Adaptive Block Processing for Multispectral Images.* IEEE Transactions on Circuits and Systems for Video Technology, 2013. **23**(9): p. 1491-1501.

15. Papathanassiadis, T. *Image block partitioning: a compression technique suitable for parallel processing*. in *1992 International Conference on Image Processing and its Applications*. 1992.
16. Usman, M.A., M.R. Usman, and S. Soo Young. *A no reference method for detection of dropped video frames in live video streaming*. in *2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN)*. 2016.
17. Lee, L.M. and F.R. Jean. *Analysis-by-synthesis frame dropping algorithm together with a novel speech recognizer using time-varying hidden Markov model*. in *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. 2014.
18. Byung-Gyu, K. and P. Dong-Jo, *Adaptive image normalisation based on block processing for enhancement of fingerprint image.* Electronics Letters, 2002. **38**(14): p. 696-698.
19. Pan, F., et al. *Proactive frame-skipping decision scheme for variable frame rate video coding*. in *2004 IEEE International Conference on Multimedia and Expo (ICME) (IEEE Cat. No.04TH8763)*. 2004.
20. Tarvainen, J., M. Nuutinen, and P. Oittinen. *Spatial and Temporal Information as Camera Parameters for Super-resolution Video*. in *2012 IEEE International Symposium on Multimedia*. 2012.
21. *https://www.youtube.com/watch?v=8aFkNRrj8n8&t=5s*.