Content Based Video Abstraction and Compression of Traffic Surveillance Videos

Shuyu Zhao, Jannatul Maksuda, Gauravi Chavan, Anup Basu and Chenqiu Zhao

Department of Computing Science,
University of Alberta, Canada
{szhao5,mourey,gauravi,basu,chenqiu1}@ualberta.ca

Abstract. The promising outcomes and use case of a technical system to ensure safe and efficient traffic management while dealing with automobiles is attracting law enforcement departments of smart cities. Therefore, efforts are being directed towards replacing the manual system that has long and monotonous surveillance videos demanding constant supervision, by a technical system that offers all informative details in a shortened version of the same video. Through our proposed work, we introduce a model that distinguishes the object of interest (vehicles) from its background, tracks the trajectory of each of this identified object and integrates trajectories occurring at different time instances from the original video into a series of frames playing simultaneously in the summarized abstract video. We have tested our proposed system by conducting experimental results on a test video that was captured under daylight conditions by a still camera. The experimental results indicate the practical applicability of our proposed method.

Keywords: Trajectory Detection, Dense Trajectories, Video Indexing, Perceptual Video Summarization, Surveillance Video Abstraction, Static Video Summary, Video Skimming.

1 Introduction

We are certain that a problem of this caliber, which directly addresses the essence of Computer Vision to automate a task that the human visual system struggles do, will attract the folks and researchers working in Computer Vision. Therefore, we are keen on proposing a model that generates an abstract and summarized traffic surveillance video. Moreover, a solution to overcome this problem will also solve similar issues faced by the officials monitoring surveillance videos in hospitals, old-age homes, etc on daily basis.

There is no denying in the fact that the supervision of any surveillance video is not only a tedious task but also demands constant attention which is difficult for humans. Therefore, our primary objective lies in providing a brief recap of the highlights of the original video. For accomplishing this, our idea is to create a short version of a traffic video that would combine trajectories of vehicles occurring at different time instances from the original video and to arrange such trajectories in a sequence of frames happening at the same time in the video's shortened version. This shortened version of the original video will serve as our finished product.

Our proposed model performs background subtraction, trajectory detection and integration using original traffic surveillance videos to produce abstract videos. If the officials employed to monitor surveillance videos use these abstracted videos, they would still be able to take note of important events occurring in the entire run time of the original video. This would make their tiring task easier. Besides, we have introduced a new algorithm for integrating trajectories simultaneously in a small portion of the abstracted video.

The rest of the paper is organized in six sections. The first section states the general overview of our research work. In section two, we mention the previous researchers' work that we have reviewed. In section three we discuss our proposed framework, functionality and construction of our "Trajectory Detection and Augmentation Model" in detail. In section four we demonstrate our results, graphs and comparative analysis. Section five outlines the factors that can be incorporated in future whereas section six summarizes our research work in brief.

2 Related Work

In [1] the authors present a trajectory clustering method for video surveillance and monitoring systems. The clusters are dynamic and built-in real-time as the trajectory data is acquired, without the need for any offline processing step. The authors also show how the obtained clusters can be successfully used for both-giving proper feedback to the low-level tracking system and collecting valuable information for the high-level event analysis modules. In this paper, the authors have used three steps to do trajectory clustering: 1. finding a suitable way to represent tracks and clusters, 2. defining a distance measured between tracks and clusters, and 3. defining a cluster updating function. However, in the method proposed by us, we calculate trajectories by comparing the position of a particular vehicle in successive frames rather than clustering the trajectories.

In paper [2], the authors have introduced an approach which emphasizes on both the content balance and the perceptual quality of a summary. The normalized cut algorithm is employed to globally and optimally partition a video into clusters. A motion attention model based on human perception is employed to compute the perceptual quality of shots and clusters. These clusters, together with the computed attention values, form a temporal graph similar to Markov chain that inherently describes the evolution and perceptual importance of video clusters. The flow of a temporal graph is utilized to group similar clusters into scenes, while the attention values are used as guidelines to select appropriate sub-shots in scenes for summarization. In their application, they used the cluster to find out shots in the same place or they share similar thematic content and then using their proposed algorithm created the summarized version of the video, on the other hand in our paper we have combined two different shots is one shot provided that the trajectories of the ROI (region of interest) are different.

In the paper [3] the authors have proposed a vehicle tracking algorithm that takes a new occlusion reasoning approach. They consider two different types of occlusions: explicit occlusion and implicit occlusion. They also proposed a traffic low extraction method with the velocity and trajectory of the moving vehicles. The proposed vehicle tracking system is composed of three parts: vehicle segmentation, vehicle tracking, and traffic parameter extraction. The vehicle segmentation part separates moving vehicles from their background. They employ an adaptive background approach, which does not update the background of moving objects. They have also designed a 20 token-based algorithm for vehicle tracking using Kalman filtering that has a modest amount of computational complexity. Moreover, they have generated a summary image based on motion focussing process as opposed to our method wherein we have generated a summary video without changing its corresponding trajectories.

In [4] the researchers have proposed a novel method for detecting nonconforming trajectories of objects as they pass through a scene. Most existing methods use spatial features to solve this problem. However, using only spatial information is not adequate; as velocity and curvature information of a trajectory along with the spatial information is crucial for finding an elegant solution. Moreover, their method has the ability to distinguish between objects traversing spatially dissimilar paths, or objects traversing spatially proximate paths but having different spatio-temporal characteristics. Their method consists of a path building training phase and a testing phase. During the training phase, they used graph-cuts for clustering the trajectories, wherein the Hausdorff distance metric is used to calculate the edge weights. Each cluster represents a path in this approach. An envelope boundary and an average trajectory is computed for each path here. During the testing phase, they use three features for trajectory matching in a hierarchical fashion. The first feature measures the spatial similarity while the second feature compares the velocity characteristic of trajectories. Finally, the curvature features capture discontinuities in the velocity, acceleration, and position of the trajectory. Nevertheless, in our paper, we have introduced a new method to find similar and dissimilar trajectories of the moving objects in a scene and have combined the objects with dissimilar trajectories into a same scene.

In paper [5], the authors have proposed a smart video summarization technique that compiles the synopsis of event(s)-of-interest occurring within a segment of image frames in a video. The proposed solution space consists of extracting appropriate features that represent the dynamics of targets in surveillance environments using their motion trajectories combined with a finite state automaton model for analyzing state changes of such features to detect and localize event(s)-of-interest. The authors introduce the cumulative moving average (CMA) and the preceding segment average (PSA) statistical metric as features that indicate gradual and sudden change in the instantaneous velocity of moving targets. In order to support both online and off-line summarization, a finite state machine, that is often referred to as a Mealy Machine, has been proposed by them to model the trajectory of a moving target and used for detecting transitions that represent a change from one state to another when initiated by a triggering event or condition. While these authors use Mealy Machine, which starts its state if an object moves in the scene; our proposed model extracts the moving object from the foreground and then finds the trajectory of each of these objects, rather than combining the objects with disparate trajectories together.

In [6], the authors introduce a quantity that provides both a cost function for temporal registration and a proper distance for comparison of trajectories. This distance is used to define statistical summaries, such as sample means and covariances, of synchronized trajectories and "Gaussian-type" models to capture their variability at discrete times. It is invariant to identical time-warping (or temporal re-parameterizations) of trajectories, which is based on a novel mathematical representation of trajectories, termed as transported square-root vector field (TSRVF), and the square of L norm on the space of TSRVFs. In their approach the authors calculate the trajectory of the object based on shape and mathematical representations whereas in our proposed algorithm, we create a bounding box around the moving object to track the position of each box that helps us in trajectory detection.

In [7] an overview of the SOM-based methodology is being introduced by the authors for video summarization. This method uses temporal trajectories of the best-matching units of frame-wise feature vectors for shot boundary detection and shot similarity assessment. In our approach, we have introduced an alternative way to find the trajectory of the moving object by tracking the location of the bounding box that is created around it in each successive frame.

In [8] the researchers have introduced a new methodology for the generation of short and coherent video summaries based on clustering of similar activities. Objects with similar activities are easy to watch simultaneously, and outliers can be spotted instantly using their approach. Based on this idea, the authors of this paper have optimized their model. Although clustering is a widely used approach for video abstraction, like the one used here, we on the contrary have used trajectory detection for producing a summarized and abstracted output video.

In [9] the researchers introduce us to a system that tracks moving objects in a video dataset so as to extract the representation of these objects' 3D trajectories. This system then finds hierarchical clusters of similar trajectories from their video dataset. For trajectory-based clustering and retrieval, they use a modified version of edit distance, called longest common sub sequence. Similarities are computed between projections of trajectories on coordinate axes. Trajectories are group based, and the authors use an agglomerate clustering algorithm to accomplish their objectives. The computation of similarity and dissimilarity between trajectories is being done by them using clusters, however, we do it by comparing the indices of pairwise trajectories.

In paper [10], the authors have build on dense trajectories, a state-of-the-art approach for describing spatio-temporal patterns in video sequences. They have also demonstrated an application of these dense trajectories for detecting objects in surveillance video. This was a totally different approach which we reviewed only to evaluate its behaviour and application while monitoring surveillance videos.

In [11] order to browse the video data quickly and make full use of the surveillance video data, the authors of [11] propose a fully automatic and computationally efficient framework for analysis and memorization of surveillance videos with the techniques of moving object detection and trajectory extraction. The video is first partitioned into segments based on moving object detection, then trajectory is extracted from each moving object, and then key frames are selected, together with the trajectories to represent the video segment. Our approach is strongly influenced by this approach.

Paper [12] presents an algorithm for anomaly detection in vehicle trajectory data using hausdorff distance. This algorithm has the capability of handling non-uniform data, data of unequal length, and data on different directions. Their proposed technique identifies anomalous trajectories and also those ones that partially behave like anomalous activities. In this technique the clusters of the nearest trajectories are formed based on hausdorff distance. The outlier trajectories are identified based on user defined outlier threshold. If any cluster contains less number of trajectories than the outlier threshold, then the trajectories of those clusters are identified as outlier trajectories. Again this approach is used by the authors to compute the similarity and dissimilarity in the trajectories being computed whereas as discussed earlier we compare trajectories based on the index assigned to each trajectory.

We took clues and planned out what approach would be suitable for us by meticulously reviewing these research papers.

3 Proposed Framework - Trajectory Detection and Augmentation Model

Our problem is to detect and combine different trajectories of motion in a shorter piece of video, for reducing the total count of frames that the trajectories had occupied separately in the original video. In order to achieve this, we classify our work in three major sections as discussed below:

- Background Subtraction
- Trajectory Detection
- Combining multiple trajectories together which would play simultaneously in the resultant video

3.1 Background Subtraction

We first disintegrate our video in a series of frames and then apply on them different background subtraction methods utilizing traditional image processing techniques. Using background subtraction algorithms, we convert a binary mask to a bounding box by finding the border pixels of the masked image. We then apply contouring on each frame so that we could draw a bounding box around each detected contour(vehicle in our case). We fix a certain threshold value for the objects to be attested as vehicles, which are our objects of interest. We then specify the top left and bottom right points of the rectangle that we wish to describe around the vehicle. In this way we get a resultant background subtracted video with a bounding box moving along with the mobile vehicle. We have compared the results obtained after performing background subtraction methods such as MOG (Mixture of Gaussians), it is a Gaussian Mixture-based Background/Foreground Segmentation Algorithm, MOG2 (Mixture of Gaussians), it is also a Gaussian Mixture-based Background/Foreground Segmentation Algorithm, GMG (Geometric Multigrid), this algorithm combines statistical background image estimation and per-pixel Bayesian segmentation and KNN(K-Nearest Neighbors) on our test video. From our findings we decided to use GMG out of all the approaches as it was the most suitable one for our specific model. We mathematically explain our implementation code for background subtraction in (Fig.1).

3.2 Trajectory Detection

We find the corners of each bounding box and store it in a linked list for a certain frame say ‘A’. In one of the fields of the linked list we assign it an index say ‘z’. We then assign the same index ‘z’ for the most proximate neighbour of the bounding box that is considered in frame ‘A’. We keep doing this in a loop for consecutive frames until there exists no neighbour for the bounding box in question. This is how we obtain a trajectory for a given object. We then eliminate the detected trajectories with very brief duration (noise) as they are redundant and the chances of them being detected as a mobile vehicles is highly unlikely.

Let's say we have a video $'V'$ comprising of a $'n'$ bundle of frames from $f_1, f_2, f_3, ...f_n$.

$V = \{f_1, f_2, f_3, ...f_n\}$

Now as we describe a bounding box around our objects of interest in each of these frames, we have a set '$b1$' comprising of boxes from 1 to 'n'.

$b_1 \rightarrow box_1, box_2, box_3........box_n$

It should be noted that we are making bounding boxes based on pixel connectivity under a proximate range.

$P_1, P_2, P_3, ..., P_n$

Pixels within very close connectivity form a cluster. In this way, we have clusters from G_1 to G_n each consisting of pixels from P_1 to P_n.

$G_1, G_2, G_3, ..., G_n$

We aim in knowing how we can group a cluster in a frame by determining the starting pixel coordinates of each cluster.

$\forall P_i P_j \in G_n$ and $P_i \cap P_j = 1$

We first apply a filter on this detected cluster to get the required background subtracted mask.

$B_n \xrightarrow{filter} B_m$

Then we create bounding box in the particular frame based on the position of the pixel of each cluster group.

$G_n = \{P_i \mid (P_i \cap P_j) = 1, P_i, P_j \in G\}$

$f_1 = \{P_1(x,y) \mid x \in (1,L), y \in (1,R)\}$ *where* $x = min(P(x) \forall P \in G)$ *and* $y = min(P(y) \forall P \in G)$
-- (1)

Fig. 1. Algorithm for Background Subtraction

We mathematically represent our pseudo code for identifying trajectory of each mobile object as demonstrated in (Fig.2).

Now let's consider bounding box $'box_1'$ corresponding to cluster $'G_1'$ and $'box_2'$ corresponding to $'G_2'$. For each of these two clusters we find the top, bottom, left and right coordinates.

$$box_1 = G_1 \ \ where\ G_1 = \{P(x-y) \mid x \in [L_1, R_1],\ y \in [T_1, B_1]\}$$

$$box_2 = G_2 \ \ where\ G_2 = \{P(x-y) \mid x \in [L_2, R_2],\ y \in [T_2, B_2]\}$$

We find the mean square difference between these four coordinates and check whether this value is less than a pre-set threshold. If the mean square value is less than the threshold, we conclude that it is the trajectory of the same object. Otherwise, we do not identify it as the same trajectory.

$$\sqrt{|L_2 - L_1|^2 + |R_2 - R_1|^2 + |T_2 - T_1|^2 + |B_2 - B_1|^2} < T_G \text{ --------------------------------------(2)}$$

Fig. 2. Algorithm for Trajectory Detection

3.3 Combining two dissimilar trajectories in a sequence of frames

Lastly using a few python libraries and the core logic that we have developed, we combine multiple separate trajectories that were running at different time instances in the original video. As a result, in the abstract video these trajectories run simultaneously and thereby decrease the total run time of the entire video. We briefly explain the logic that we implemented while integrating trajectories. We combine multiple trajectories into a small portion of the video running simultaneously, so that a compressed video is obtained out of it. For this, we first obtain a static background image using background subtraction on a small clip of a video and then place non-colliding trajectories on this static background. In order to understand this, we encourage you to have a look at our combining scheme briefly outlined in (Fig.3):

To sum up, we present the work-flow of our proposed system in (Fig.4):

Using conventional math and image processing techniques we have designed a system to serve the above-mentioned purpose. We used Python, OpenCV and a few python libraries for programming our model.

We faced a number of challenges while we were trying to accomplish the milestones that we had set for this research work. A few crucial ones are discussed as under:

1. Conducting experimental analysis on test videos with different lighting conditions was a tedious task. These videos each with different lighting conditions affected the background subtraction differently. Therefore, we had to filter on the prerequisites of the video that is suitable for our model.We finally decided to go ahead with a video captured in daylight conditions.

Let's say we record the trajectory coordinates
$(t_i = \{left - L_i, right - R_i, top - T_i, bottom - B_i\}\ )$ for trajectory $'T_1'$ and
$(t_j = \{left - L_j, right - R_j, top - T_j, bottom - B_j\})$ coordinates for the trajectory $'T_2'$.

$$T_1 = \{t_i = (L_i, R_i, T_i, B_i) \mid i \in [1, N_i]\}$$

$$T_2 = \{t_j = (L_j, R_j, T_j, B_j) \mid i \in [1, N_j]\}$$

We define a function by substituting t_i and T_2 values in equation (2) which are two different trajectories. This is how we combine two different trajectories. We continue this so on so forth with all distinct trajectories. In this way, we combine multiple trajectories to form a final summarized abstract video.

$$f(\ t_i|T_2) = f(\ t_i \mid t_{i,j} \in [1, N]\ )$$

$$= \ \left|(L_i - L_j) * (L_i - R_j) < 0\right| \ \cap \ \left|(R_i - L_j) * (R_i - R_j) < 0\right| \ \cap \ \ldots\ldots\ldots\ldots = 1 \ \textit{then do not combine}$$

Fig. 3. Algorithm for combining multiple dissimilar trajectories in a sequence of frames

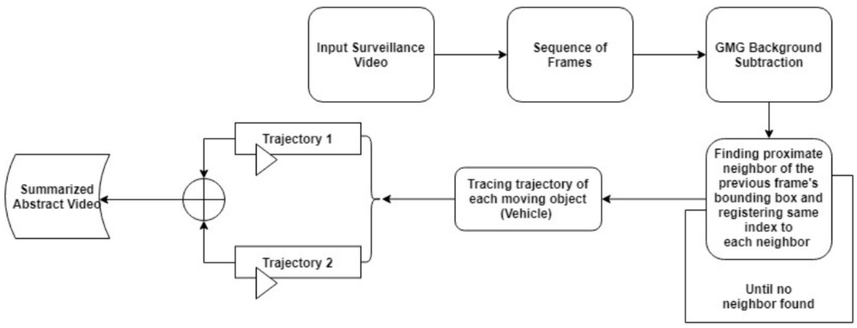

Fig. 4. Work flow of Our Proposed Research Work

2. There were instances wherein the vehicle's bounding box would disappear as the vehicles approached towards the end of the camera view. This happened because the bounding box exceeded its limit to the point where there was no neighbour available for it.
3. At the time of writing this paper, there weren't any references or past work for simultaneously combining two separate trajectories in a series of frames. Therefore, it was a challenging task to come up with an idea for implementing this fresh thought.

It's also important to discuss what sets our research work apart from the rest and what are we offering to the research sector that hasn't been yet introduced. Out of the available approaches such as 'feature-based', 'event-based', 'shot selection-based', 'cluster-based', 'trajectory-based' and 'mosaic-based' for video abstraction; we chose to go ahead with trajectory-based approach for identifying and combining trajectories of mobile objects. We did so after reviewing the above mentioned approaches and analyzing their outcomes. The trajectory-based approach was befitting for the use case that we had planned for our product design. Out of all the remaining approaches, since only a few researchers have worked on the traffic management use case using trajectory detection at the time of writing this paper, we thought of delving deep into this idea.

Moreover, we have proposed a new model for video abstraction/summarization, wherein we introduce a novel technique to integrate separate trajectories occurring at different time instances from the original video into the resultant abstracted video. As a result, these trajectories occur simultaneously in the resultant video.

4 Experimental Results

We demonstrate the outcomes of this research work in this section. We also discuss in detail the results of the comparative analysis we conducted while evaluating the usability, agility, feasibility and resourcefulness of our proposed system. We have investigated the outcomes of four background subtraction methodologies that are MOG, MOG2, KNN and GMG. These results are of these four techniques are demonstrated in (Fig.5).

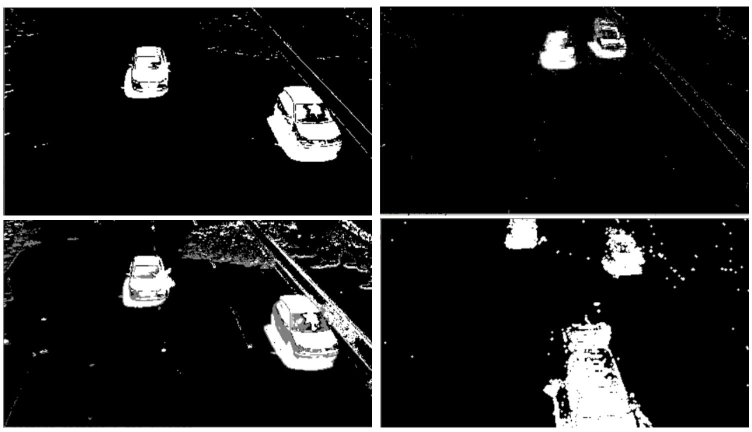

Fig. 5. Result of four background subtraction methodologies that are MOG (top left), MOG2 (top right) ,KNN (bottom left) and GMG (bottom right)

Although KNN (bottom left in Fig. 5) showed the most promising results for background subtraction, while implementing trajectory detection we realized that GMG (bottom right in Fig. 6) is the best suited and feasible approach for our model.

In the figure (Fig.6), we outline our step-wise implementation wherein the first sub section demonstrates the bounding box around our subjects of interest(vehicles). The second sub section displays a green bounding box in the current frame that is a neighbour of the white bounding box in the previous frame. The third picture shows how we track the trajectories of individual vehicles whereas the final one eliminates the redundant noise in the video (trajectories with brief duration).

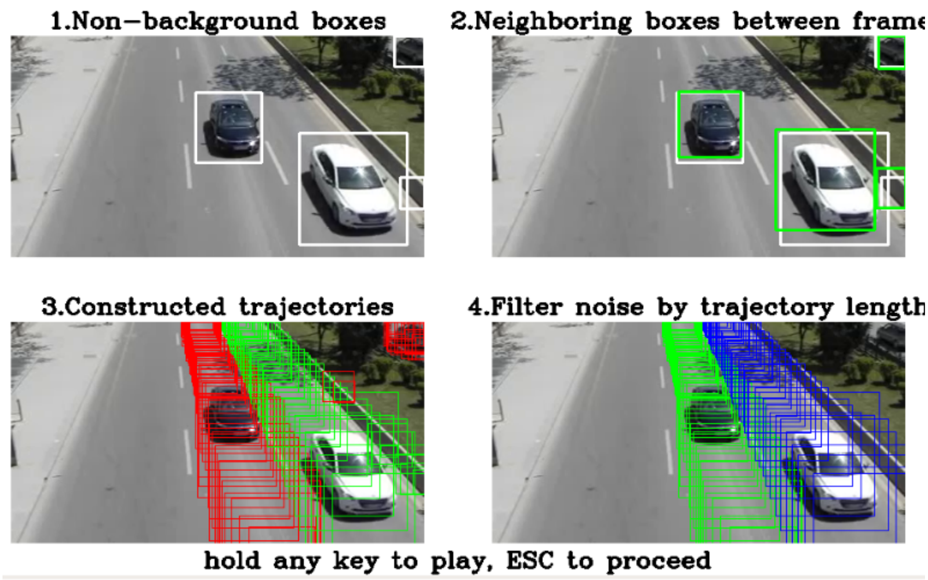

Fig. 6. Step-Wise Implementation of Our Proposed Model

We compared the outcomes of our original video and the resultant summarized video after using our modeled algorithm. As shown in figure (Fig.7), we can clearly observe that after using our proposed algorithm the number of frames in the original video and its size have notably reduced.

We have tested our proposed system using a test video which was captured under daylight conditions by a still camera.

5 Future Scope

1. Incorporating a relatively high level abstraction such as adding and synchronizing sound of individual objects occurring at different time instances

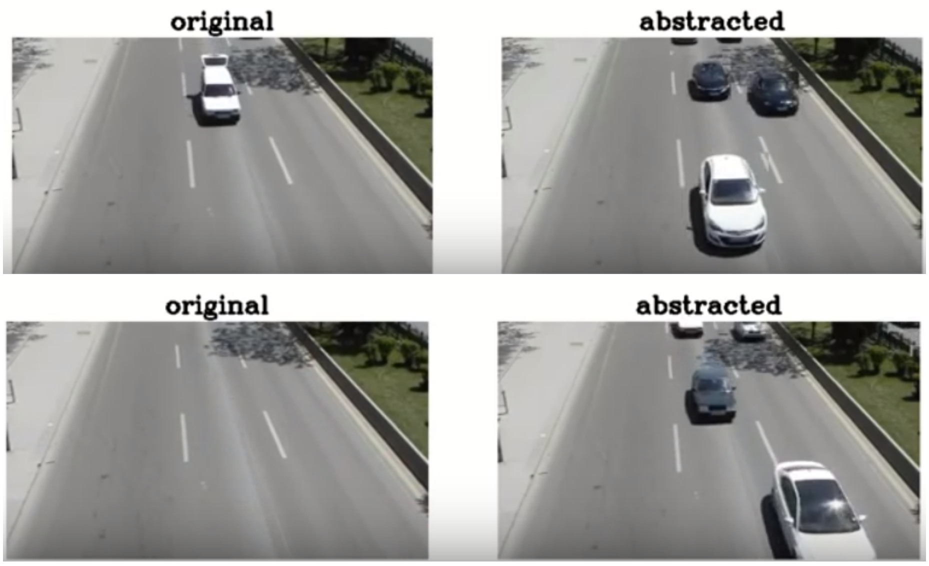

Fig. 7. Comparison between the Original Video Frame and Abstract Video Frame that combines different objects in a single frame after using Our Proposed Algorithm

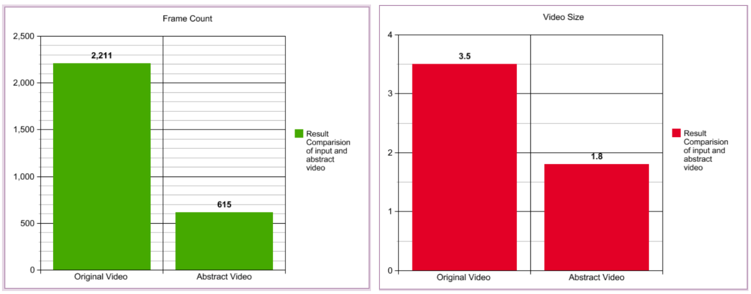

Fig. 8. Comparison between input and abstract video based on total count of frames (left side) and video size (right side)

from the original surveillance video into a chunk of the abstracted video, wherein the trajectories would be playing simultaneously leaves room for further research.

2. Detecting trajectories of vehicles under poor lighting conditions could be a challenging task for the researchers, as it would need a suitable scheme to adjust such conditions using Computer Vision enhancement techniques.
3. Although we have designed this model to detect and augment trajectories of individual objects that are captured by a single still camera; the repercussions of the footage obtained by multiple camera angles of the same scene and

their impact on trajectory detection and augmentation could be investigated further.
4. The velocities of the trajectories of individual objects can be computed in future. This will allow researchers to eliminate the trajectories of unwanted moving objects (stray animals), moving across/along the road. The direction where a vehicle is headed to could also be estimated once the scheme to find velocity is introduced.

6 Conclusion

The objective of this research work is to develop a summarized abstract video which recaps the important events from the long surveillance input video. We have developed this model for the traffic management use case, wherein we distinguish our object of interest (vehicle) from its background, track the trajectory of each detected vehicle and combine different trajectories occurring at different time instances from the original video into a series of frames playing simultaneously in the summarized abstract video. This would reduce the total count of frames that the trajectories had occupied separately in the original video.

Our work flow has three major sections where we first perform background subtraction to describe a bounding box around each moving vehicle. Out of the many approaches that we investigated, we found the GMG background subtraction method to be most suitable for our model. We then found the most proximate neighbour in the current frame of the bounding box from the previous frame and assign both a common index. We continue this process until there exists no neighbour for the bounding box in question. This is how we detect a trajectory for each vehicle. Lastly, using a few python libraries and the core logic that we have developed, we combined multiple separate trajectories from the original video and synchronized them to play simultaneously in the resultant abstract video.

The novelty of our project lies in the section wherein we combine multiple trajectories together in a series of frames of the resultant video. To the best of our knowledge no one has implemented this in a similar capacity so far (till the time of publishing this paper).

We have tested our proposed system by conducting experimental results on a test video that was captured under daylight conditions by a still camera. Moreover, the experimental results indicate the practical applicability of our proposed method.

References

1. L Snidaro C Piciarelli, GL Foresti. Trajectory clustering and its applications for video surveillance. 2005.

2. Hong-Jiang Zhang Chong-Wah Ngo, Yu-Fei Ma. Automatic video summarization by graph modeling. 2003.
3. Shiaw-Shian Yu Tsuhan Chen Congcong Li, Yi-Ta Wu. Motion-focusing key frame extraction and video summarization for lane surveillance system. 2009.
4. George Kollios Dan Buzan, Stan Sclaroff. Extraction and clustering of motion trajectories in video. 2004.
5. Harish Bhaskar Debi Prosad Dogra, Arif Ahmed. Smart video summarization using mealy machine-based trajectory modelling for surveillance applications. 2015.
6. Eric Klassen Jingyong Su, Sebastian Kurtek and Anuj Srivastava. Statistical analysis of trajectories on riemannian manifolds: Bird migration, hurricane tracking and video surveillance. 2014.
7. Mats Sjoberg Ville Viitaniemi Jorma Laaksonen, Markus Koskela and Hannes Muurinen. Video summarization with soms. 2007.
8. Young-Kee Jung and Yo-Sung Ho. Traffic parameter extraction using video-based vehicle tracking. 1999.
9. Member Sinnu Susan Thomas, Sumana Gupta and Venkatesh K. Subramanian. Event detection on roads using perceptual video summarization. 2018.
10. Shubhrata Gupta Vaishali Mirge, Kesari Verma. Outlier detection in vehicle trajectories. 2000.
11. Avishai Hendel Yael Pritch, Sarit Ratovitch and Shmuel Peleg. Clustered synopsis of surveillance video. 2009.
12. Rongrong Qian Jintao Ma Zhong Ji, Yuting Su. Surveillance video summarization based on moving object detection and trajectory extraction. 2010.